AC7 File Format Specification

Author: https://github.com/michgz
Date: 18-Feb-2024
Document Version: 1.1

Introduction

This is an unofficial formal specification of the AC7 file format used by Casio keyboards to store rhythm data. Casio has never released a specification for this format so all the information here has been pieced together through trial-and-error by members of the Casio user community.

The format specified is for the CT-X3000 and CT-X5000 models and their regional variants. Other Casio models may implement the format in ways which differ from what is described here.

Formatting

Numbers written in Roman font are in decimal format. Numbers written in **Courier** font are in hexadecimal format.

32B = 32 bits
32B-LE = little-endian 32-bit number
16B-LE = little-endian 16-bit number

Example

127 and **07F** are the same number.

Definitions

Part

This is a term used by Casio to refer to the data assigned to a single synthesiser channel. Each part has one instrument assigned to it at a time, and also has a channel in the on-keyboard GUI mixer. Each rhythm has 8 parts, as defined in the table:

Index	Casio name	Yamaha name	Instruments supported	Chord sync/chord substitution supported
1	Percussion	Percussion	Drum kits only	None

2	Drum	Rhythm		
3	Bass	Bass	All	All
4	Chord 1	Chord 1	All	Most (bass-specific “f-root” chord substitutions excluded)
5	Chord 2	Chord 2		
6	Chord 3	Pad		
7	Chord 4	Phrase 1		
8	Chord 5	Phrase 2		

Yamaha names are included above as a guide to possible usage – for example, Chord 3 part may often be used for a Pad sound. It’s not a requirement to use it in that way, and in fact you have a lot of flexibility in how to use the parts musically.

Element

This is a term used by Casio to refer to a time segment of music data within a rhythm. Each rhythm has 12 elements, with 10 of them accessible from a CT-X5000 keyboard. They are listed in the table:

Index	Casio Name	Notes
1	Intro	
2	Variation 1	
3	Variation 2	
4	Fill 1	
5	Fill 2	
6	Ending	
7		Not accessible – all tracks in this element should be empty.
8	Variation 3	
9	Variation 4	
10	Fill 3	
11	Fill 4	
12		Not accessible – all tracks in this element should be empty.

The “Variation” elements loop endlessly until another element is selected, while the Intro, Ending and Fill elements usually play once only.

Part/Element Combination (PEC)

This is a term defined here. It refers to a unique combination of a part and an element in a rhythm. With 8 parts and 12 elements, there are 8 x 12 = 96 part/element combinations.

Track

This is a term defined here. It refers to a sequence of music data (equivalent to a track in a Type II MIDI file) which has Chord sync and Chord substitution defined on it and is associated with a Part/element combination.

Each part/element combination has at least one track associated to it. Most commonly it will have exactly one – the most common exceptions to that rule are outlined in ?Ref?. Therefore most rhythms will contain exactly 96 tracks, although some may contain a few more than that.

Empty track

A track with no “Note” events (i.e. no music).

Atom

A term defined here. It is a variable-length data structure that is widely used by Casio. The format is:

Byte 1: value 0–255, indicating the type of atom
Byte 2: value 0–255, indicating the length of the “Payload” portion.
Variable number of bytes (0 or more): the “Payload” portion.

Examples:

An atom with no payload:

```
FF 00
```

An atom with two bytes of payload:

```
11 02 0A 23
```

Chord Sync

A Casio term – Yamaha uses the term “Transposition” for the same meaning.

Chord sync can be either “On” or “Off”. If “On”, the rhythm track data is assumed to be in C major and it is transposed to the requested pitch level. For example, if the user plays a D minor chord, then the track will be transposed up 2 semitones (to D *major* – the change to D minor happens under Chord conversion not Chord sync!)

If “Off”, no transposition is done – the rhythm track data is used as it is. This is the only possible behaviour for tracks in parts Percussion and Drum. Also, no subsequent Chord conversion is done if Chord sync if Off.

Chord Conversion

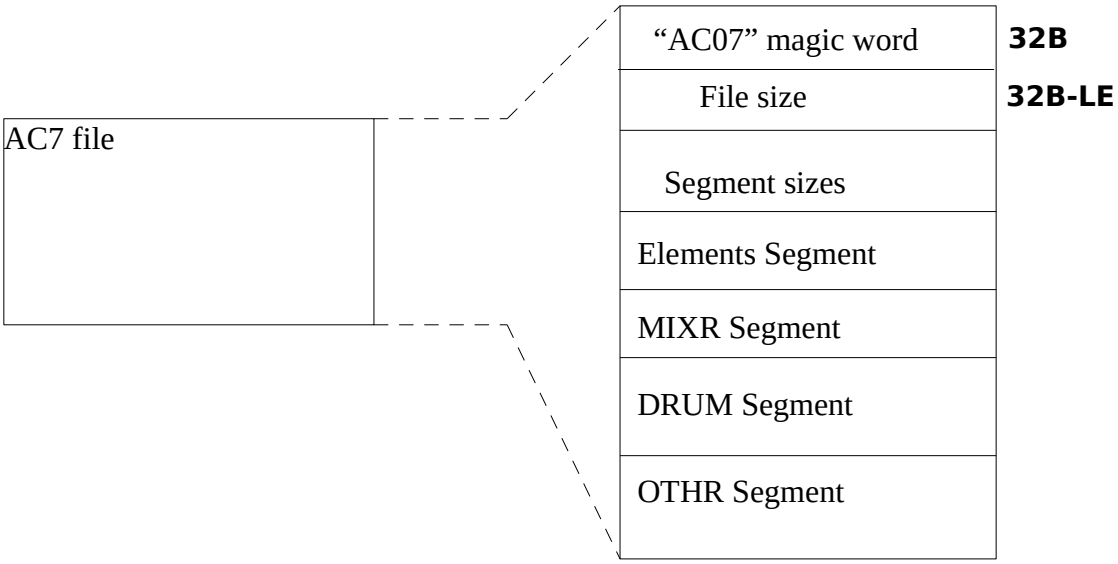
A Casio term. Specifies how note conversions are done based on the type of chord that a user has played.

Must be one of 36 different values. 18 of them are documented in the CT-X5000 manual, while the remaining ones are used in built-in rhythms but are not publicly documented.

Number Value	Name	Description
0	Bass Basic	
1	Bass 7th	
2	Chord Basic	
3	Chord Var2	
4	Chord Var3	
5	Chord Var4	
6	Chord 7th	
7	Chord Minor	
8	Phrase	
9	Chord Minor Bass	
10	Penta Phrase	
11	Intro natural-minor	
12	Intro melodic-minor	
13	Intro harmonic-minor	

...etc…

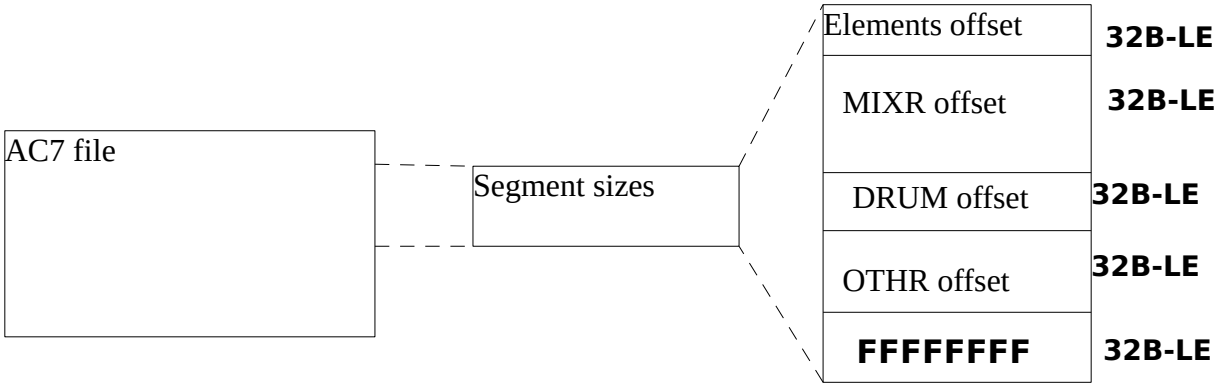
Overall structure



“AC07” magic word	32 bit value, consisting of the ASCII characters ‘A’ ‘C’ ‘0’ ‘7’.
File size	32-bit value, the full size of the AC7 file.

b’A’ b’C’ b’0’ b’7’	4 character bytes	Special value to indicate the start of a AC7 rhythm file
File Size	32 bit value, little-endian	Total size in bytes of the AC7 file – from before the start of “AC07” to the last byte of OTHR data.

Segment sizes



Name	Meaning	Format
Elements offset	Offset of Elements segment, in bytes starting from beginning of file	32 bits, little-endian
MIXR offset	Offset of ‘MIXR’ segment, in bytes starting from beginning of file	32 bits, little-endian
DRUM offset	Offset of ‘DRUM’ segment, in bytes starting from beginning of file	32 bits, little-endian
OTHR offset	Offset of ‘OTHR’ segment, in bytes starting from beginning of file	32 bits, little-endian
FFFFFFFF	End indication (fixed value)	32 bits, little-endian

Elements segment

Magic word **07FFFFFF**	Indicates start of segment; value **07FFFFFF**	32 bit little-endian

Segment size	Length of the segment in bytes	16 bits little-endian
Element count	Number of elements; always value 12	8 bits
Element Definition Offset 1	Offset in bytes of the first Element Definition from the start of this segment	32 bit little-endian
Element Definition Offset 2	Offset in bytes of the 2nd Element Definition from the start of this segment	32 bit little-endian
...	...	...
Element Definition Offset 12	Offset in bytes of the last Element Definition from the start of this segment	32 bit little-endian
00 Atom	Rhythm Name	12-byte payload; ASCII terminated with Null (**00**) character and then padded to 12 bytes with any characters
01 Atom	Time Signature. Not used?? (Element-specific values used instead)	1 byte payload (3 bytes total)
02 Atom	Tempo in BPM. Not used?? (User selected value used instead)	1 byte payload (3 bytes total)
09 Atom	Volume	1 byte payload (3 bytes total)
40 Atom	Reverb type	1 byte payload (3 bytes total)
46 Atom (optional; multiple)	Reverb parameters	
41 Atom	Chorus type	
47 Atom (optional; multiple)	Chorus parameters	
42 Atom	Delay type	
48 Atom (optional; multiple)	Delay parameters	
45 Atom (optional; multiple)	Delay parameters	
11 Atom (optional; multiple)	Rhythm button allocation; associates front-panel buttons to elements	2 byte payload (4 bytes total)
FF atom	End indication	0 byte payload (2 bytes total)
Element Definition 1		Many bytes; see below for format
Element Definition 2		Many bytes; see below for format
….	….	….
Element Definition 12		Many bytes; see below for format

Element definition

Magic word ‘ELMT’	‘E’, ‘L’, ‘M’, ‘T’	32 bits
Length	Length of this definition in bytes, starting from start of ‘ELMT’ magic word	16 bits, little endian
01 atom	Time signature	1 byte payload (3 bytes total)
06 atom	Number of measures	1 byte payload (3 bytes total)
07 atom	Number of tracks	1 byte payload (3 bytes total)
20 atom	Track index (in DRUM/OTHR) for each track	2 bytes per track
21 atom	Mixer index for each track	2 bytes per track
22 atom	Part indicator for each track	1 byte per track
30 atom (optional)	Delay send for each part	8 byte payload (10 bytes total)
FD atom		0 byte payload (2 bytes total)
36 atom (optional, multiple)		
FE atom		0 byte payload (2 bytes total)
31 atom (optional, multiple)		
32 atom (optional, multiple)		
35 atom (optional, multiple)		
FF atom	End indicator	0 byte payload (2 bytes total)

Ordering of tracks

Atoms 20, 21 and 22 contain information on each of the tracks associated with the element. They are ordered first by part, then by track.

For example, if each PEC for this element contains only one track, then atoms 20, 21 and 22 will each contain 8 data items ordered as:

Part1, part 2, part 3, part 4, part 5, part 6, part 7, part 8.

If Part 3 has two tracks and all the others have 1, then there will be 9 data items ordered as:

Part1, part 2, part 3 first track, part 3 second track, part 4, part 5, part 6, part 7, part 8.

Part Indicators

These are 1 byte per track, with the following values:

Value	Part	Track data
0F	1 (Percussion)	In ‘DRUM’ segment
00	2 (Drum)	In ‘DRUM’ segment
01	3 (Bass)	In ‘OTHR’ segment
02	4 (Chord 1)	In ‘OTHR’ segment
03	5 (Chord 2)	In ‘OTHR’ segment
04	6 (Chord 3)	In ‘OTHR’ segment
05	7 (Chord 4)	In ‘OTHR’ segment
06	8 (Chord 5)	In ‘OTHR’ segment

Some values can be added to change the behaviour of the track:

Added value	Name	Effect
A0	Minor Only	This track is only sounded when accompaniment is a “Minor-type” chord – e.g. m, m7, dim7 etc.
80	Major Only	This track is only sounded when accompaniment is *not* a “Minor-type” chord – that is, it is a “Major-type” chord such as maj7,Mmin7, aug, etc – or if there is no accompaniment.
10	No Chord-Sync	There is no Chord Sync/Chord Substitution applied to this track. Only valid for parts 3-8 – parts 1 & 2 never have Chord Sync applied.

Track Indices

These are the offset index of the track data within either the ‘DRUM’ segment or the ‘OTHR’ segment (as determined according to the Part Indicator, as above).

A value of **8000** is added to each index.

Mixer Indices

These are the offset index of the mixer data within the ‘MIXR’ segment. Only the first data item for a part contains valid data. If a PEC has more than one track, the first track for that part will have the mixer offset index and subsequent data items will have an invalid value of **FFFF**.

A value of **8000** is added to each valid index.

Mixer segment

The Mixer segment defines mixer settings for each Part/Element Combination (of which there are 96).

b’M’ b’I’ b’X’ b’R’	4 character bytes	Special value to indicate the start of the Mixer segment
Mixer Size	32 bit value, little-endian	Size in bytes of the Mixer segment – from start of “MIXR” to the end of Mixer data
n	16 bit value, little-endian	Number of entries in the Mixer data – always 96.
Addr 1	32 bit value, little-endian	Address of Mixer Data 1, from the start of the AC7 file
Addr 2	32 bit value, little-endian	Address of Mixer Data 2, from the start of the AC7 file
:		:
Addr *n*	32 bit value, little-endian	Address of Mixer Data *n*, from the start of the AC7 file
Mixer Data 1	6 bytes	See below for format
Mixer Data 2	6 bytes	See below for format
:		
Mixer Data *n*	6 bytes	See below for format

With a value of n=96, the total size of this segment will be **3CA** bytes.

Mixer Data

Each of the 96 mixer data entries has the following format:

Byte Index	Name	Format
0	Patch	1 byte, values 0–127
1	Bank MSB	1 byte, values 0–120
2	Volume	1 byte, values 0–127
3	Pan	1 byte, values 0–127. Centre is 64.
4	Reverb effect send	1 byte, values 0–127
5	Chorus effect send	1 byte, values 0–127

Drum Segment

b’D’ b’R’ b’U’ b’M’	4 character bytes	Special value to indicate the start of the Drum segment
Drum Size	32 bit value, little-endian	Size in bytes of the Drum segment – from start of “DRUM” to the end of Drum data
n	16 bit value, little-endian	Number of entries in the Drum data.
Addr 1	32 bit value, little-endian	Address of Drum Data 1, from the start of the AC7 file
Addr 2	32 bit value, little-endian	Address of Drum Data 2, from the start of the AC7 file
:		:
Addr *n*	32 bit value, little-endian	Address of Drum Data *n*, from the start of the AC7 file
Drum Track Data 1		See below for format
Drum Track Data 2		See below for format
:		
Drum Track Data *n*		See below for format

Other Segment

b’O’ b’T’ b’H’ b’R’	4 character bytes	Special value to indicate the start of the Othr segment
Othr Size	32 bit value, little-endian	Size in bytes of the Othr segment – from start of “OTHR” to the end of Othr data
n	16 bit value, little-endian	Number of entries in the Othr data.
Addr 1	32 bit value, little-endian	Address of Othr Data 1, from the start of the AC7 file
Addr 2	32 bit value, little-endian	Address of Othr Data 2, from the start of the AC7 file
:		:
Addr *n*	32 bit value, little-endian	Address of Othr Data *n*, from the start of the AC7 file
Othr Track Starter 1	3 bytes	See below for format
Othr Track Data 1		See below for format
Othr Track Starter 2	3 bytes	See below for format
Othr Track Data 2		See below for format
:		
Othr Track Starter *n*	3 bytes	See below for format
Othr Track Data *n*		See below for format

Starter

The starter is a 3-byte value which precedes the track data on every track in a OTHR segment. It is 3 bytes long and contains information on Chord Sync settings.

It is a bit-packed structure of six fields:

Chord conversion	Break point	Inversion	Retrigger	F-root	Lowest note
23 16	15 12	11 9	8	7	6 0

Chord Conversion: 8 bits. Takes values 0-35 as specified above in the Definitions section.

Break point: 4 bits. Takes values 0-11, corresponding to notes C-B (0=C, 1=C#, 2=D, …, 11=B).

Inversion: 3 bits. Takes value 0=”Off”, 2=”On”, 4=”7th”. Other values are no used.

Retrigger: 1 bit. Takes values 0=”On”, 1=”Off”. (Note these are the opposite way round from expected!)

F-Root: 1 bit. Takes values 0=”Off”, 1=”On”.

Lowest note: 7 bits. Takes values 0-127, corresponding to MIDI Note Number of the lowest allowed note after chord conversion.

Track data

Track data has the same format in the DRUM and OTHR segments.

It consists of 3-byte events concatenated in time order. The last event must always be a End-Of-Track.

An event is:

1 Byte: *tt*
1 Byte: *nn*
1 Byte: *vv*

Byte Index	Name	Description
0	*tt*	Time delta. The time between the preceding event and this one. Units are 96 per quarter note.
1	*nn*	See below. Often MIDI note pitch
2	*vv*	See below. Often MIDI note-on velocity

tt is the time delta. It's the time between the previous event (or track start time if the first event) and the current event. Units are 96 per quarter note.

The remaining two bytes are interpreted according to the following table:

nn Values	*vv* Values		
0–127	1–127	Note On	Equivalent to a MIDI Note On event with Note Number *nn* and Velocity *vv*.
0–127	0	Note Off	Equivalent to a MIDI Note Off event with Note Number *nn* and Velocity 127. (Note-Off velocity is not alterable).
8E	0–255	Pitch bend	
B0	0–127	Modulation (Vibrato Depth)	Equivalent to a MIDI Controller #1 change command (Modulation) with data *vv*.
B1	0–127	Assignable control	Can be assigned to control a Biquad filter or DSP parameter. Only available in built-in rhythms, no way to use in a User rhythm.
B5	0–127	Expression (Volume)	Equivalent to a MIDI Controller #11 change command (Expression) with data *vv*.
B9	0–10	Pitch Bend Range	
BA	0–127	Filter Cut-Off	Equivalent to a MIDI Controller #74 change command (Filter Cut-Off) with data *vv*.
BB	0–127	Filter Resonanace	Equivalent to a MIDI Controller #71 change command (Filter Resonance) with data *vv*.
BC	0–127	Attack Time	Equivalent to a MIDI Controller #73 change command (Attack Time) with data *vv*.
BD	0–127	Release Time	Equivalent to a MIDI Controller #72 change command (Release Time) with data *vv*.
E0	0–31	Chord	Changes to a new table from

		Conversion table	the one selected in the track starter
E1	0–2	Chord Inversion selection	Changes to a new selection from the one in the track starter
E2	0, 2	Chord Retrigger setting	Changes to a new setting from the one in the track starter
E3	0–255	Tempo Fast	Adds *vv* bpm to the base tempo
E4	0–255	Tempo Slow	Subtracts *vv* bpm from the base tempo
E5	0	Allow User Editing	If this is the first event in a track, allows the user to edit the track in the CT-X5000 interface. Otherwise, attempting to edit the track will result in it being cleared to an empty track first. There's usually no reason to *not* include this event. It should be the first event in every track except in elements 7 & 12.
E6	0–127	Chord Conversion Highest Note	Sets the highest note value of all notes after chord sync.
E7	0–127	Chord Conversion Pitch Hint	Hint for the pitch at change of chord (often used for Bass part?)
FC	0	End-Of-Track	Must be the last event in any track
FF	0–255	Time Jump	Creates a time delta which may be larger than 255 ticks. Additional jump of 255×*vv* ticks – so the total time delta from this event will be tt + 255×*vv*. A special case when tt=128 and vv=4: this jumps to the end time of the track, no matter how long the track is.

Examples

An empty track will have the following formats:

Elements 7 & 12:

80 FF 04 00 FC 00

All other elements:

00 E5 00 80 FF 04 00 FC 00

Track-to-PEC matching

As mentioned above, there are a few (unusual) cases where more than one track is associated with a PEC. There must always be at least one track associated to a PEC.
Broadly, two or more tracks are used when two different Chord Sync/Chord Substitution settings must be applied to the data. Common cases where this might be true are:

- There is a difference of behaviour between Major & Minor chords. Then one track might use a “Major Only” Chord Sync setting and the other a “Minor Only”.
- A Versatile instrument is selected. Chord Sync should be applied to notes up to B6, while the “special effects” notes from C7 up should have no Chord Sync.
- The Orchestra Set drum kit is selected. Chord Sync should be applied to the Timpani (melodic) notes, but not to any other notes. (Similarly for Yun Luo sounds from the Chinese Set).

Selected Atom Formats

The format of certain atom types is defined here, with more possibly being added over time. Note that the CT-X specific atoms **`31`**, **`32`**, **`33`**, **`35`** and **`36`** are defined in the Documentation folder of the ac7maker project.

Note that meaning or validity of any atom is dependent on where it is used in the AC7 file.

Time Signature (`01`)

A 1-byte payload specifying a time signature. Only a few specific values are allowed; the complete list of them are shown in the table:

Time Signature	Payload value
2/4	**12**
3/4	**1A**
4/4	**22**
5/4	**2A**
6/4	**32**
7/4	**3A**
8/4	**42**
2/8	**13**
3/8	**1B**
4/8	**23**
5/8	**2B**
6/8	**33**
7/8	**3B**
8/8	**43**
9/8	**4B**
10/8	**53**
11/8	**5B**
12/8	**63**
13/8	**6B**
14/8	**73**
15/8	**7B**
16/8	**83**

Delay Send (30)

An 8-byte payload. Each byte takes values 0-127, and the meanings are:

Byte 0: Delay send for Part 1
Byte 1: Delay send for Part 2
Byte 2: Delay send for Part 3
Byte 3: Delay send for Part 4
Byte 4: Delay send for Part 5
Byte 5: Delay send for Part 6
Byte 6: Delay send for Part 7
Byte 7: Delay send for Part 8

Front-Panel Key Association (11)

Associates some of the VARIATION/FILL buttons on the keyboard front-panel with rhythm elements.

2-byte payload. The first payload byte defines the button and takes one of the following values:

Value of Byte 0	Meaning
06	No meaning?
07	VARIATION 3 on CT-X3000/5000
08	VARIATION 4 on CT-X3000/5000
09	FILL 3 on CT-X3000/5000
0A	FILL 4 on CT-X3000/5000
0B	No meaning?

Note that only buttons which are specific to the CT-X3000/5000 are defined. On a CT-X700/800 keyboard this atom will have no effect.

The second payload byte defines a rhythm element and takes one of the following values:

Value of Byte 1	Meaning
00	Intro
01	Unknown?
10	Variation 1
11	Variation 2
12	Variation 3
13	Variation 4
20	Fill 1
21	Fill 2
22	Fill 3
23	Fill 4
30	Ending
31	Unknown?

All rhythms created on a CT-X3000/5000 keyboard contain the following 6 atoms at the relevant place in the Elements Segment. It’s recommended that for a rhythm to work correctly on those keyboards, these 6 atoms be included exactly in this way:

11 02 06 01
11 02 07 12
11 02 08 13
11 02 09 22
11 02 0A 23
11 02 0B 31