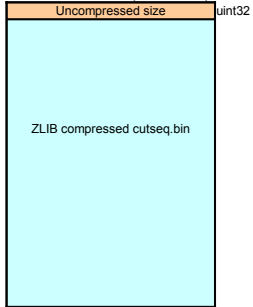


Tomb Raider Cutseq File Format
Document Version: 1

uint32 = unsigned 32 bit integer
sint32 = signed 32 bit integer

Number of cutscenes:
TR4: 30 (n = 1 to n = 30)
TR5: 44 (n = 1 to n = 44)
TIMES: 4 (n = 28 to n = 31)
TIMES: Offset & Size for directory entries 1 to 27 are 0.

CUTSEQ.PAK FILE (TR4, TIMES)



Python 2.5 code to decompress:

```
# import the zlib library
import zlib

# cutseq.pak in same folder as python script
infile = open("cutseq.pak", "rb")

# read 4 bytes
size = infile.read(4)

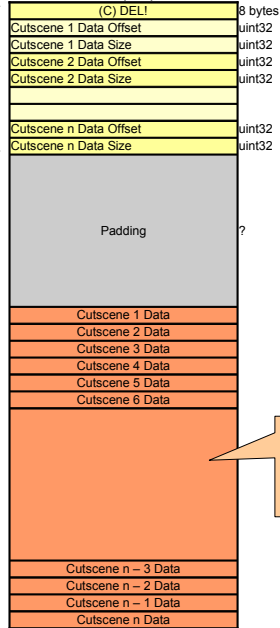
# read the compressed data
comp_data = infile.read()

# close the file
infile.close()

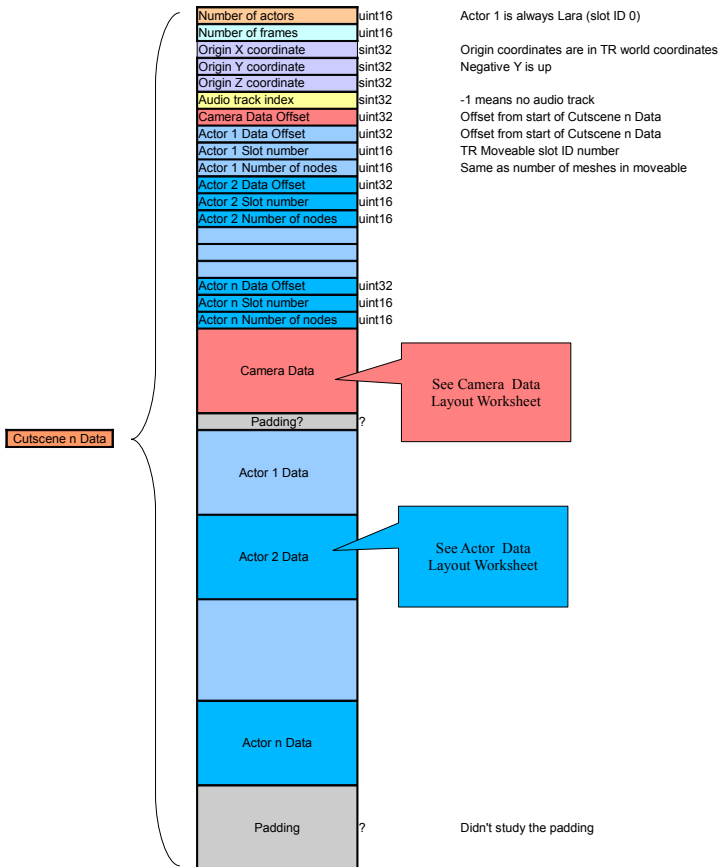
# decompress the data
uncomp_data = zlib.decompress(comp_data)

# write uncompressed data to file
outfile = open("cutseq.tmp", "wb")
outfile.write(uncomp_data)
outfile.close()
```

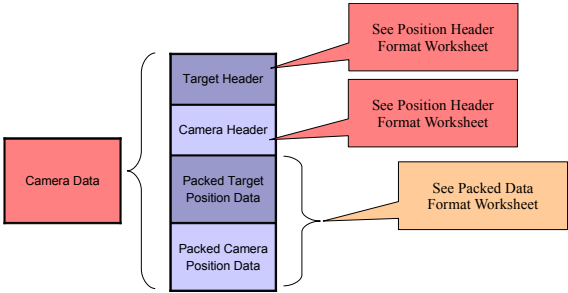
CUTSEQ.BIN FILE (TR5)

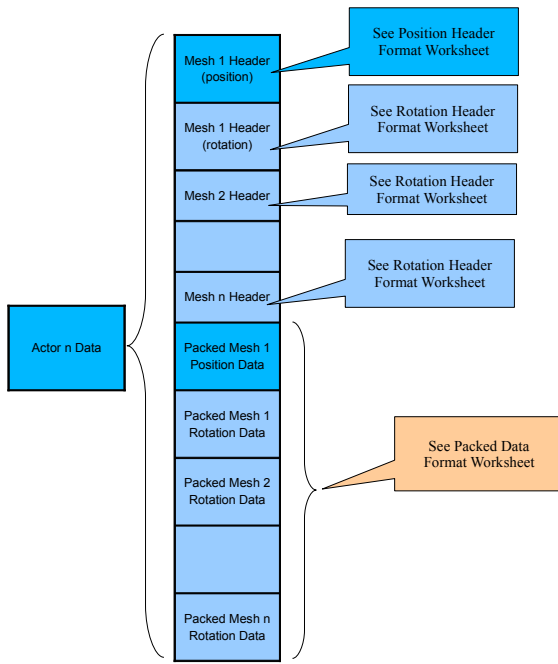


Cutscene Data Format



Camera Data Layout





Position Header Format

Coordinates are relative to Cutscene Origin

Start position X coordinate	sint16	
Start position Y coordinate	sint16	
Start position Z coordinate	sint16	
Bitsizes for each axis (X, Y, Z)	uint16	Bitfield, [b16,b15,b14, ..., b2,b1]
Number of values packed in X axis data	uint16	Each value is (X axis bitsize) bits long
Number of values packed in Y axis data	uint16	Each value is (Y axis bitsize) bits long
Number of values packed in Z axis data	uint16	Each value is (Z axis bitsize) bits long

Decoding axis bitsizes:		courtesy T4Larson
X axis bitsize = b14 b13 b12 b11		(Bitsizes >> 10) & 0x0f
Y axis bitsize = b9 b8 b7 b6		(Bitsizes >> 5) & 0x0f
Z axis bitsize = b4 b3 b2 b1		(Bitsizes & 0x0f)
		Minimum bitsize is 6

		Python code
--	--	-------------

Rotation Header Format

1024 = 360 degrees

Start rotation about X axis	sint16
Start rotation about Y axis	sint16
Start rotation about Z axis	sint16
Bitsizes for each axis (X, Y, Z)	uint16
Number of values packed in X axis data	uint16
Number of values packed in Y axis data	uint16
Number of values packed in Z axis data	uint16

Bitfield, [b16,b15,b14, ..., b2,b1]
 Each value is (X axis bitsize) bits long
 Each value is (Y axis bitsize) bits long
 Each value is (Z axis bitsize) bits long

Decoding axis bitsizes:

X axis bitsize = b14 b13 b12 b11

Y axis bitsize = b9 b8 b7 b6

Z axis bitsize = b4 b3 b2 b1

courtesy T4Larson

(Bitsizes >> 10) & 0x0f

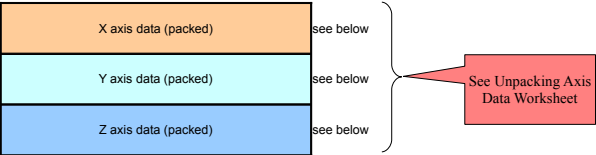
(Bitsizes >> 5) & 0x0f

(Bitsizes & 0x0f)

Minimum bitsize is 6

} Python code

Packed Position and Rotation Data format



Length of packed axis data (number of bytes):
 $((\text{axis bitsize} * \text{number of values in axis data}) / 8) + 4$

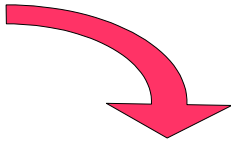
courtesy T4Larson
integer division

Unpacking Axis Data

Data is from TR4 cutscene 28, Actor 2, Mesh 1, X, Y & Z position data and Mesh 1, X, Y & Z rotation data.

Bitsize = 6
Number of values = 12
Data length = 13 bytes

Packed Axis Data	0xd7
	0x07
	0x5c
	0x1f
	0x70
	0x7d
	0x40
	0x34
	0x00
	0x00
	0x00
	0x00



Reversed for demonstration

0x00	0x00	0x00	0x00	0x00	0x34	0x40	0x7d	0x70	0x1f	0x5c	0x07	0xd7
------	------	------	------	------	------	------	------	------	------	------	------	------

Binary

0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0011 0100	0100 0000	0111 1101	0111 0000	0001 1111	0101 1100	0000 0111	1101 0111
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------

12 groups of 6 bits
(number of values of bitsize)

0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0000 0000	0011 0100	0100 0000	0111 1101	0111 0000	0001 1111	0101 1100	0000 0111	1101 0111
					last value							first value	

12 values in correct order

01 0111	01 1111	00 0000	01 0111	01 1111	00 0000	01 0111	01 1111	00 0000	01 0001	00 0011	00 0000
---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------	---------

If most significant bit is 1, bitwise OR the value with bitmask.

Bitsize	6	7	8	9	10	11	12	13	14	15
Mask	0xff c0	0xff 80	0xff 00	0xfe 00	0xfc 00	0xf8 00	0xf0 00	0xe0 00	0xc0 00	0x80 00

Unpacked values

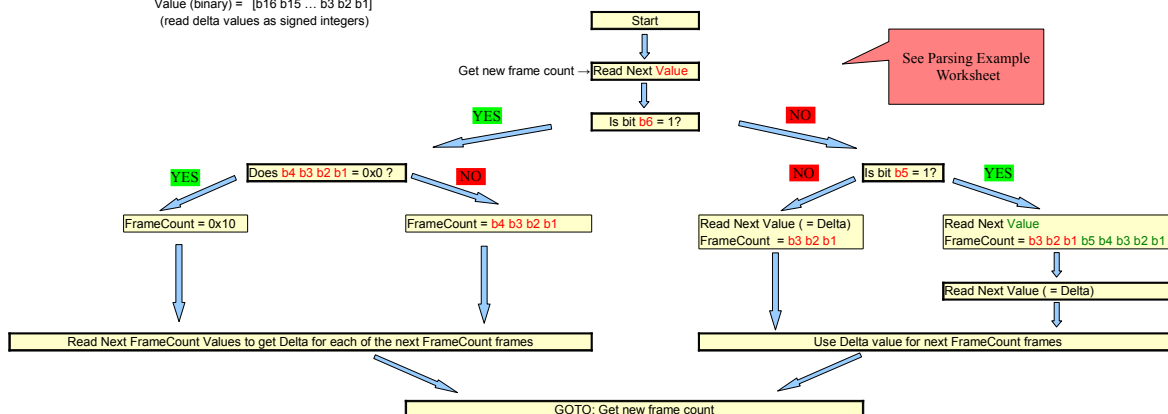
1	2	3	4	5	6	7	8	9	10	11	12
0x17	0x1f	0x00	0x17	0x1f	0x00	0x17	0x1f	0x00	0x11	0x03	0x00

See Parsing Unpacked Axis Data Worksheet

Parsing Unpacked Axis Data

Flowchart:

Value (binary) = [b16 b15 ... b3 b2 b1]
(read delta values as signed integers)



Python 2.5 code:

```

# fc is framecount, v is list of unpacked values
# the delta value for each frame is added to a list
while len(v) > 0:
    # while there are unread values
    fc = v.pop(0)
    # pop first value
    flag1 = (fc >> 5) & 0x1
    if flag1:
        # bit6 = 1
        if fc & 0x0f == 0:
            fc = 0x10
        else:
            fc = fc & 0x0f
        for i in range(fc):
            # for framecount frames
            deltas.append(v.pop(0))
            # add delta read from list
        else:
            flag2 = (fc >> 4) & 0x1
            if flag2:
                # bit5 = 1
                fc1 = v.pop(0)
                # read next value
                delta = v.pop(0)
                # read next value
                fc = (fc & 0x7) << 5
                fc1 = (fc1 & 0x1f)
                fc = fc | fc1
            else:
                # bit5 = 0
                delta = v.pop(0)
                # read next value
                fc = (fc & 0x7)
                for i in range(fc):
                    # for framecount frames
                    deltas.append(delta)
                    # add same delta to list
    
```

The packed axis data thus determines a delta value for each frame of the cutscene and the delta values are used as follows.

Before the cutscene begins, the position or rotation values (x,y,z) of the mesh are set the same as the start position or rotation values.

The delta values (x,y,z) for frame 1 are then added to the position or rotation values to give the position or rotation for frame 1.

The delta values (x,y,z) for frame 2 are then added to the frame 1 position or rotation values to give the position or rotation for frame 2.

and so on.

Notes:

For rotation the addition is mod 1024.

The bitsizes for the packed data vary due to the magnitude of the largest delta value.

I only studied the TRLR tomb4.exe where it read the cutseq file so there may be further processing of the extracted data in other routines.

By trial and error I found that for Moveables the position (x,y,z) of the root mesh for each frame must be multiplied by 3 for the animations to be correct.

T4Larson later confirmed that there was a routine in the code that did this.

T4Larson also found in the code that the position (x,y,z) of the camera and camera target for each frame must be multiplied by 2.

Parsing Example

This data is for the root mesh (Mesh 1) of the pedestal (Actor 2) in cutscene 28 where Lara reveals the hidden skull in the TR4 title.
The root mesh does not change position and does not rotate.
The cutscene is 800 frames long.

Unpacked values

Binary
[b8 b7 b6 b5 b4 b3 b2 b1]

1	2	3	4	5	6	7	8	9	10	11	12
0x17	0x1f	0x00	0x17	0x1f	0x00	0x17	0x1f	0x00	0x11	0x03	0x00
0001 0111	0001 1111	0000 0000	0001 0111	0001 1111	0000 0000	0001 0111	0001 1111	0000 0000	0001 0001	0000 0011	0000 0000

Read value 1 (Get framecount)	0x17	
b6 = 0		
b5 = 1		
Read value 2	0x1f	
FrameCount (binary/decimal)	1111 1111	255 (255 appears to be the maximum allowable framecount)
Read value 3 (= Delta)	0	
Summary	0 is the delta value for the first 255 frames	
Read value 4 (Get framecount)	0x17	
b6 = 0		
b5 = 1		
Read value 5	0x1f	
FrameCount (binary/decimal)	1111 1111	255
Read value 6 (= Delta)	0	
Summary	0 is the delta value for the next 255 frames	
Read value 7 (Get framecount)	0x17	
b6 = 0		
b5 = 1		
Read value 8	0x1f	
FrameCount (binary/decimal)	1111 1111	255
Read value 9 (= Delta)	0	
Summary	0 is the delta value for the next 255 frames	
Read value 10 (Get framecount)	0x11	
b6 = 0		
b5 = 1		
Read value 11	0x03	
FrameCount (binary/decimal)	0010 0011	35
Read value 12 (= Delta)	0	
Summary	0 is the delta value for the next 35 frames	
Total number of delta values =	255 + 255 + 255 + 35 =	800
Number of frames in cutscene =	800	

So for each frame we have a delta value of 0 to add to the current position/rotation to obtain the new position/rotation.
The root mesh therefore stays in it's starting position and pose.

List of TR4 cutscenes

Some cutscenes have entries in the script with the command Cut or ResidentCut.

Cut is for cutscenes at the beginning of the level.

ResidentCut is for cutscenes that use a Cutscene Trigger in the level.

For those cutscenes that do not have script entries I deduced the level they occur in by matching the slot numbers for the actors listed for that cutscene.

Cutscene #	TR4 File	Level Name	Description	Script Entry
1	joby1a	The Sphinx Complex	Lara pulling out shovel and starting to dig	ResidentCut
2	joby1a	The Sphinx Complex	Lara standing and putting shovel away	ResidentCut
3	joby1a	The Sphinx Complex	Lara kneeling, reading inscription then standing	ResidentCut
4	joby3a	Menkaure's Pyramid		ResidentCut
5	angkor1	Angkor Wat	Young Lara getting backpack	ResidentCut
6	angkor1	Angkor Wat	Young Lara appearing from behind Von Croy at level start	ResidentCut & Cut ??
7	ang_race	Race For The Iris	Young Lara winning race	ResidentCut
8	ang_race	Race For The Iris	Von Croy winning race	ResidentCut
9	ang_race	Race For The Iris	Revealing Iris, Von Croy getting trapped and Young Lara going for help	None
10	libend	Hall Of Demetrius		None
11	highstr1	Citadel Gate		None
12	karnak1	Temple Of Karnak	Lara arriving in jeep	Cut
13	hall	The Great Hypostyle Hall		None
14	settlomb2	Burial Chambers	Lara pulling Ankh off Seth's Tomb	ResidentCut
15	citnew	Citadel		None
16	citnew	Citadel		None
17	alexhub	Alexandria		ResidentCut
18	alexhub	Alexandria		ResidentCut
19	alexhub	Alexandria		ResidentCut
20	alexhub	Alexandria		ResidentCut
21	alexhub2	Coastal Ruins	Lara using pole and hook to get keys	ResidentCut
22	palaces2	Cleopatra's Palaces	Lara approaching and sitting on throne	ResidentCut
23	joby5b	Temple Of Horus		ResidentCut
24	lowstr1	Street Bazaar		None
25	highstr1	Citadel Gate		None
26	highstr1	Citadel Gate		None
27	joby5c	Temple Of Horus		ResidentCut
28	title	Title	Lara revealing hidden golden skull	ResidentCut
29	title	Title	Lara crossing falling wooden bridge	ResidentCut
30	title	Title	Lara fighting Mummy	ResidentCut

TIMES LEVEL:

Cutscene #	TR4 File	Level Name	Description	Script Entry
28	title	Title	Lara revealing hidden golden skull	ResidentCut
29	title	Title	Lara crossing falling wooden bridge	ResidentCut
30	title	Title	Lara fighting Mummy	ResidentCut
31	office	The Times Office	Lara talking to TIMES CEO and then going to archive	Cut

TR4 cutscene information output by my exploratory program.

DataDump function by T4Larson

Data for Actor 1, cutscene 28 extracted

Note I use different terminology. Tracks are the packed axis position or axis rotation data. Actor 0 is the first actor. Mesh 0 is the first mesh.

output format						
Cutscene #	Cutscene Offset	(Cutscene size)	#A: Num actors	#F: Num frames	(Origin)	Audio track
Actor number	Actor data offset	(Size Actor data)	[Num meshes]	Moveable slot		Cam data offset
1. 0x00001000 (0x1fe8) #A: 1 #F: 257 (51109, 256, 50995) : 47 : 32						
0. 0x00000230 (7608) [15] : LARA						
2. 0x00003000 (0x10c4) #A: 1 #F: 151 (51109, 256, 50995) : -1 : 32						
0. 0x0000014c (3960) [15] : LARA						
3. 0x00004800 (0x7885) #A: 1 #F: 1942 (50783, 26, 53612) : 92 : 32						
0. 0x00001c44 (23617) [15] : LARA						
4. 0x0000c800 (0xc29f) #A: 4 #F: 900 (66404, -7680, 44551) : 81 : 56						
0. 0x00000bb4 (13024) [15] : LARA						
1. 0x00003e94 (14500) [15] : TROOPS						
2. 0x00007738 (16048) [27] : SCORPION						
3. 0x0000b5e8 (3255) [5] : SETHA_MIP						
5. 0x00019000 (0xc250) #A: 3 #F: 1750 (100843, 768, 58873) : 89 : 48						
0. 0x000018bc (27820) [15] : LARA						
1. 0x00008568 (1252) [2] : JEAN_YVES_MIP						
2. 0x00008a4c (14340) [21] : CROCODILE_MIP						
6. 0x00025800 (0x2912) #A: 2 #F: 430 (5632, 1280, 86528) : 48 : 40						
0. 0x0000051c (6352) [15] : LARA						
1. 0x00001dec (2854) [25] : VON_CROY						
7. 0x00028800 (0x40e7) #A: 3 #F: 880 (52736, -3330, 49664) : 82 : 48						
0. 0x00000bd4 (6392) [15] : LARA						
1. 0x000024cc (6180) [25] : VON_CROY						
2. 0x00003cf0 (1015) [13] : MUTANT_MIP						
8. 0x0002d000 (0x5e5b) #A: 3 #F: 710 (52736, -3330, 49664) : 76 : 48						
0. 0x00000d28 (14736) [15] : LARA						
1. 0x000046b8 (5188) [25] : VON_CROY						
2. 0x00005afc (863) [13] : MUTANT_MIP						
9. 0x00033000 (0x22a7c) #A: 10 #F: 3330 (52736, -3330, 49664) : 100 : 104						
0. 0x00004c38 (44204) [15] : LARA						
1. 0x0000f8e4 (61448) [25] : VON_CROY						
2. 0x0001e8ec (11980) [13] : MUTANT_MIP						
3. 0x000217b8 (476) [1] : BADDY_1_MIP						
4. 0x00021994 (336) [1] : BADDY_2_MIP						
5. 0x00021ae4 (364) [1] : HORSEMAN_MIP						
6. 0x00021c50 (336) [1] : SCORPION_MIP						
7. 0x00021da0 (472) [1] : JEAN_YVES_MIP						
8. 0x00021f78 (2128) [8] : TROOPS_MIP						
9. 0x000227c8 (692) [4] : KNIGHTS_TEMPLAR_MIP						
10. 0x00056000 (0x175d7) #A: 6 #F: 2650 (70073, -2304, 59833) : 96 : 72						
0. 0x000028d8 (28516) [15] : LARA						
1. 0x0000983c (25196) [25] : ANIMATING1						
2. 0x0000faa8 (13496) [21] : SPHINX_MIP						
3. 0x00012860 (8056) [21] : SPHINX_MIP						
4. 0x00014ed8 (9664) [21] : SPHINX_MIP						
5. 0x00017498 (319) [1] : DOOR_TYPE5						
11. 0x0006d800 (0x2e1a2) #A: 7 #F: 3400 (54784, -1792, 76260) : 103 : 80						
0. 0x00005328 (73444) [15] : LARA						
1. 0x0001720c (9032) [11] : MOTORBIKE						
2. 0x00019554 (35168) [15] : ANIMATING8						
3. 0x00021eb4 (32972) [28] : MUTANT						
4. 0x00029f80 (14684) [13] : ANIMATING6						
5. 0x0002d8dc (1968) [5] : ANIMATING4						
6. 0x0002e08c (278) [1] : ANIMATING7						
12. 0x0009c000 (0x624b) #A: 2 #F: 750 (89529, 0, 46553) : 74 : 40						
0. 0x00001234 (14920) [15] : LARA						
1. 0x00004c7c (5583) [18] : ANIMATING6						
13. 0x000a2800 (0x7ab7) #A: 8 #F: 800 (81408, 0, 86528) : 72 : 88						
0. 0x00000bf4 (868) [15] : LARA						
1. 0x00000f58 (6728) [25] : SETHA_MIP						
2. 0x000029a0 (4032) [21] : KNIGHTS_TEMPLAR_MIP						
3. 0x00003960 (4664) [21] : KNIGHTS_TEMPLAR_MIP						
4. 0x00004b98 (4872) [21] : KNIGHTS_TEMPLAR_MIP						
5. 0x00005ea0 (3588) [18] : HORSE_MIP						
6. 0x00006ca4 (2092) [18] : HORSE_MIP						
7. 0x000074d0 (1511) [6] : MUTANT_MIP						
14. 0x000aa800 (0x5a78) #A: 2 #F: 650 (26112, 256, 64000) : 71 : 40						
0. 0x0000082c (20580) [15] : LARA						
1. 0x00005890 (488) [1] : SKELETON_MIP						
15. 0x000b0800 (0x1aee4) #A: 9 #F: 2200 (72192, 2048, 64000) : 95 : 96						
0. 0x00002680 (33540) [15] : LARA						
1. 0x0000a984 (13376) [25] : ANIMATING9_MIP						
2. 0x0000ddc4 (20596) [24] : ANIMATING8_MIP						
3. 0x00012e38 (10060) [24] : ANIMATING8_MIP						
4. 0x00015584 (9616) [24] : ANIMATING8_MIP						
5. 0x00017b14 (10600) [24] : ANIMATING8_MIP						
6. 0x0001a47c (344) [3] : ANIMATING12_MIP						
7. 0x0001a5d4 (772) [1] : ANIMATING11_MIP						
8. 0x0001a8d8 (1548) [1] : ANIMATING10_MIP						
16. 0x000cb800 (0x20a5e) #A: 10 #F: 3150 (60928, 0, 45568) : 99 : 104						
0. 0x000044b0 (17276) [15] : LARA						
1. 0x0000882c (47108) [25] : ANIMATING9_MIP						
2. 0x00014030 (18848) [21] : ANIMATING15_MIP						
3. 0x000189d0 (15260) [19] : ANIMATING14_MIP						
4. 0x0001c56c (4668) [16] : KNIGHTS_TEMPLAR						
5. 0x0001d7a8 (4088) [16] : KNIGHTS_TEMPLAR						
6. 0x0001e7a0 (536) [3] : ANIMATING12_MIP						
7. 0x0001e9b8 (3064) [1] : ANIMATING11_MIP						
8. 0x0001f5b0 (1824) [1] : ANIMATING10_MIP						
9. 0x0001fcd0 (3470) [1] : ANIMATING13_MIP						
17. 0x000ec800 (0x5885) #A: 3 #F: 800 (54784, -8192, 50688) : 73 : 48						
0. 0x00000c7c (8212) [15] : LARA						
1. 0x00002c90 (10612) [19] : JEAN_YVES						
2. 0x00005604 (641) [2] : ANIMATING4						
18. 0x000f2800 (0x4c77) #A: 2 #F: 670 (54784, -8192, 50688) : 77 : 40						
0. 0x000009cc (5108) [15] : LARA						
1. 0x00001dc0 (11959) [19] : JEAN_YVES						
19. 0x000f7800 (0x7d9b) #A: 2 #F: 1440 (54784, -8192, 50688) : 87 : 40						
0. 0x0000124c (14760) [15] : LARA						
1. 0x00004bf4 (12711) [19] : JEAN_YVES						

20. 0x000f800 (0x67a1) #A: 3 #F: 980 (54784, -8192, 50688) : 83 : 48
 0. 0x000009e8 (9336) [15] : LARA
 1. 0x00002e60 (14320) [19] : JEAN_YVES
 2. 0x00006650 (337) [2] : ANIMATING4
 21. 0x00106000 (0x3c87) #A: 2 #F: 565 (33280, -4352, 40448) : 66 : 40
 0. 0x0000072c (12840) [15] : LARA
 1. 0x00003954 (819) [1] : ANIMATING10
 22. 0x0010a000 (0xbefd) #A: 5 #F: 1000 (72192, -11776, 62976) : 84 : 64
 0. 0x0000120c (26360) [15] : LARA
 1. 0x00007904 (7716) [22] : DEMIGOD3
 2. 0x00009728 (8952) [22] : DEMIGOD3
 3. 0x0000ba20 (212) [1] : DOOR_TYPE6
 4. 0x0000baf4 (1033) [1] : DOOR_TYPE6
 23. 0x00116000 (0x3d8e) #A: 1 #F: 660 (17920, 29952, 15872) : 78 : 32
 0. 0x000009b4 (13242) [15] : LARA
 24. 0x0011a000 (0xb10a) #A: 3 #F: 2180 (53760, 0, 47616) : 93 : 48
 0. 0x00001ba0 (11232) [15] : LARA
 1. 0x00004780 (25572) [15] : ANIMATING13
 2. 0x0000ab64 (1446) [1] : ANIMATING15_MIP
 25. 0x00125800 (0xce6c) #A: 2 #F: 3400 (54784, -2304, 76288) : 101 : 40
 0. 0x00001938 (25828) [15] : LARA
 1. 0x00007e1c (20560) [15] : ANIMATING8
 26. 0x00132800 (0x266c) #A: 2 #F: 450 (54784, -2304, 76288) : 60 : 40
 0. 0x00000410 (4620) [15] : LARA
 1. 0x0000161c (4176) [15] : ANIMATING8
 27. 0x00135000 (0x4973) #A: 2 #F: 429 (16896, 17920, 15872) : 50 : 40
 0. 0x00000604 (16380) [15] : LARA
 1. 0x00004600 (883) [12] : ANIMATING5
 28. 0x0013a000 (0x31d4) #A: 2 #F: 800 (80384, -1024, 90624) : -1 : 40

Camera

Node #0: (target)

Position: (1590, 158, -467)

bitsizes: (7, 6, 6)

track #0 [619]: 04 7a a8 6e a0 87 ea 06 7a a8 7e a0 87 ea 06 7a a8 6e a0 87 ea 06 7a a8 6e a0 87 ea 05 3a 08 7f c7 df ef f8 3b fe 8e bf e3 ef 77 fc 1d 84 bf e3 ef f8 fb 1d 7f c7 df f1 77 fc fd 0e c2 df f1
 track #1 [471]: 20 cf f7 3c df f3 7c cf f3 3d df f3 3c d8 f3 7c cf f3 3d cf f7 3c df f3 7c Of f2 3d cf f7 3c df f3 7c cf f3 3d cf 83 3d cf f7 7c cf f3 3d cf f7 3c df f3 2b df f3 7c cf f3 3d df f3 d2 03 a0 bf ff ff fe fb c3 af fa ff ef ff bf ff ff fe
 track #2 [619]: e1 30 0c a4 30 0c c2 30 84 c2 30 84 c2 30 90 c2 30 08 c3 10 0a c3 40 0a c3 20 0c 43 28 0c 03 29 0c 83 30 0c a1 30 0c a1 30 0c a4 30 0c c2 30 84 c2 30 90 c2 30 08 c3 10 0a d0 02 84 3f c4 00 e1 0f 31 40

Node #1: (camera)

Position: (5434, -106, -570)

bitsizes: (7, 6, 6)

track #0 [659]: 10 04 37 d4 25 70 4d 5d 6e 97 db e5 1e ba 24 6f d7 db 2d 78 45 5f 6f e1 2b 02 7f 0f 60 51 18 4c 08 9b c4 61 b1 78 2c 1e 8c 0b e4 41 99 60 1e 9c 55 e8 33 1a 9d 46 a7 d3 ea f4 40 2d 5a ab 07 6b 11 7b 3d f
 track #1 [575]: 43 10 02 44 10 02 43 00 02 41 00 04 01 10 04 40 00 04 40 00 98 01 10 00 40 30 00 61 00 29 40 f8 0f 40 fa 03 3f 0f 03 ff 0f 40 c8 7f fa ff ef ff fe 3f f8 e1 4f f8 61 3f f8 66 ef f7 7e ef 0f 7d e8 43 49 1f f2 44 0f f2
 track #2 [640]: e1 3f f8 e1 3f f8 ec ef fb bf ff fb bf ff ff 3e 94 fc 24 10 03 ff 00 80 3f 10 00 40 00 04 01 10 08 41 20 04 84 00 0e c2 20 0c 03 31 10 03 41 14 44 41 9c 85 51 14 c6 61 0c 47 89 1c 09 82 10 89 a1 8c bc a2 0c 8b c

0. 0x00000b7c (9584) [15] : LARA
 1. 0x000030ec (232) [2] : ANIMATING12

Actor #1

Mesh0 (node #0):

Position: (0, -255, -171)

bitsizes: (6, 6, 6)

	start position		
	bitsizes for each axis		
	[number of packed values]		packed data (hex)
track #0 [12]: d7 07 5c 1f 70 7d 40 34 00 00 00 00 00	x		
track #1 [12]: d7 07 5c 1f 70 7d 40 34 00 00 00 00 00	y		
track #2 [12]: d7 07 5c 1f 70 7d 40 34 00 00 00 00 00	z		

Mesh0 (node #1):

Rotation: (0, 0, 0)

bitsizes: (6, 6, 6)

track #0 [12]: d7 07 5c 1f 70 7d 40 34 00 00 00 00 00
 track #1 [12]: d7 07 5c 1f 70 7d 40 34 00 00 00 00 00
 track #2 [12]: d7 07 5c 1f 70 7d 40 34 00 00 00 00 00

Mesh1 (node #2):

Rotation: (0, 768, 512)

bitsizes: (6, 6, 6)

track #0 [12]: d7 07 5c 1f 70 7d 40 34 00 00 00 00 00
 track #1 [12]: d7 07 5c 1f 70 7d 40 34 00 00 00 00 00
 track #2 [110]: 16 06 80 fa ae ef bb be ef fa be eb fb ae ef e0 ae ef fa be eb fb ae ef fa be eb bb 0e ee bb be eb fb ae ef fa be eb bb be 83 bb be eb fb ae ef fa ae ef bb be eb ef be eb bb be

29. 0x0013d800 (0x3f80) #A: 3 #F: 650 (40448, 0, 80384) : -1 : 48
 0. 0x00000964 (11036) [15] : LARA
 1. 0x00003480 (2376) [18] : ANIMATING16
 2. 0x00003dc8 (440) [9] : ANIMATING5
 30. 0x00141800 (0x6df5) #A: 2 #F: 750 (6656, 0, 79360) : -1 : 40
 0. 0x00000aac (11468) [15] : LARA
 1. 0x00003778 (13949) [20] : ANIMATING9

List of TR5 cutscenes

Some cutscenes have entries in the script with the command Cut or ResidentCut.

Cut is for cutscenes at the beginning of the level.

ResidentCut is for cutscenes that use a Cutscene Trigger in the level.

For those cutscenes that do not have script entries deduce the level they occur in by matching the slot numbers for the actors listed for that cutscene.

Cutscene #	TRC File	Level Name	Description	Script Entry
1				None
2	joby3	The submarine		ResidentCut
3				None
4				None
5				None
6				None
7				None
8				None
9				None
10				None
11				None
12				None
13				None
14				None
15				None
16				None
17				None
18	joby3	The submarine		Cut
19				None
20				None
21				None
22				None
23				None
24				None
25				None
26	richcut2	Security breach		ResidentCut & Cut ??
27				None
28	title	Title	Lara climbing shelves and jumping on bald enemy	ResidentCut
29	title	Title	Lara evading guard in office. Fells him with kick	ResidentCut
30	title	Title	Lara being chased by guard and laser guard	ResidentCut
31	title	Title	Lara dropping from ceiling and pushing guard into fire	ResidentCut
32	joby5	Sinking submarine		Cut
33				None
34	andy2	Labyrinth		ResidentCut
35	andy3	Old mill		Cut
36				None
37				None
38				None
39				None
40				None
41				None
42				None
43				None
44				None

TR5 cutscene information output by my exploratory program.

DataDump function by T4Larson

Data for Actor 0, cutscene 28 extracted

Note I use different terminology. Tracks are the packed axis position or axis rotation data. Actor 0 is the first actor. Mesh 0 is the first mesh.

Note invalid data offset for Actor 0 (Lara) in cutscenes 11, 26 & 40

output format							
Cutscene #	Cutscene Offset	(Cutscene size)	#A :Num actors	#F: Num frames	(Origin)	Audio track	Cam data offset
Actor number	Actor data offset	(Size Actor data)	[Num meshes]	Moveable slot			
1. 0x00000800 (0x396d) #A: 2 #F: 219 (26919, -3072, 43499) : 19 : 40							
0. 0x0000022c (8028) [15] : LARA							
1. 0x00002188 (6117) [15] : 37							
2. 0x00004800 (0x360a) #A: 2 #F: 219 (58880, 0, 47616) : 20 : 40							
0. 0x000001ec (6060) [15] : LARA							
1. 0x00001998 (7282) [15] : 416							
3. 0x00008000 (0x734f) #A: 2 #F: 349 (26919, -3072, 43499) : 43 : 40							
0. 0x00000430 (13644) [15] : LARA							
1. 0x0000397c (14803) [15] : 416							
4. 0x0000f800 (0x395c) #A: 2 #F: 218 (26919, -3072, 43499) : 30 : 40							
0. 0x0000028c (7852) [15] : LARA							
1. 0x00002138 (6180) [15] : 39							
5. 0x001a7000 (0xedc4) #A: 6 #F: 2055 (52736, 4864, 79360) : 110 : 72							
0. 0x0000205c (20500) [15] : LARA							
1. 0x00007070 (28492) [17] : 36							
2. 0x0000dfbc (1652) [10] : 42							
3. 0x0000e630 (1368) [7] : 60							
4. 0x0000eb88 (268) [1] : 54							
5. 0x0000ec94 (304) [1] : 48							
6. 0x0019c000 (0xa92c) #A: 3 #F: 1230 (47616, 7680, 26112) : 93 : 48							
0. 0x00001360 (7568) [15] : LARA							
1. 0x000030f0 (30640) [25] : 32							
2. 0x0000a8a0 (140) [1] : 34							
7. 0x00187000 (0x14864) #A: 5 #F: 1426 (55808, 0, 30208) : 98 : 64							
0. 0x000016f0 (18200) [15] : LARA							
1. 0x00005e08 (20112) [15] : 45							
2. 0x0000ac98 (15780) [12] : 63							
3. 0x0000ea3c (13352) [12] : 63							
4. 0x00011e64 (10752) [12] : 63							
8. 0x0017c000 (0xae6b) #A: 2 #F: 1538 (55808, 0, 30208) : 102 : 40							
0. 0x000017d8 (12252) [15] : LARA							
1. 0x000047b4 (26434) [15] : 45							
9. 0x0016c000 (0xfa9e) #A: 7 #F: 1143 (43520, 5376, 49664) : 89 : 80							
0. 0x00001328 (29744) [15] : LARA							
1. 0x00008758 (6956) [11] : 89							
2. 0x0000a284 (7160) [11] : 89							
3. 0x0000be7c (6568) [11] : 89							
4. 0x0000d824 (7972) [11] : 89							
5. 0x0000f748 (176) [1] : 60							
6. 0x0000f7f8 (678) [1] : 32							
10. 0x00159800 (0x12412) #A: 6 #F: 1032 (43520, 5376, 49664) : 85 : 72							
0. 0x00001184 (34828) [15] : LARA							
1. 0x00009990 (9108) [11] : 89							
2. 0x0000bd24 (7176) [11] : 89							
3. 0x0000d92c (9092) [11] : 89							
4. 0x0000fcb0 (9964) [11] : 89							
5. 0x0001239c (118) [1] : 60							
11. 0x00155000 (0x4174) #A: 3 #F: 630 (36352, 1792, 33280) : 68 : 48							
0. 0xffffffff (-4294965515) [15] : 65535							
1. 0x000006f4 (13508) [15] : 39							
2. 0x00003bb8 (1468) [6] : 56							
12. 0x00142000 (0x12e4e) #A: 2 #F: 3455 (59904, 512, 49664) : 122 : 40							
0. 0x00001fd8 (43096) [15] : LARA							
1. 0x0000c830 (26142) [23] : 56							
13. 0x00132800 (0xf1dc) #A: 3 #F: 2550 (59904, 512, 49664) : 116 : 48							
0. 0x000015e4 (30572) [15] : LARA							
1. 0x00008d50 (23992) [23] : 56							
2. 0x0000eb08 (1748) [1] : 40							
14. 0x0012a800 (0x7db9) #A: 10 #F: 942 (55808, 8448, 53760) : 81 : 104							
0. 0x00000c80 (24004) [15] : LARA							
1. 0x00006a44 (936) [7] : 38							
2. 0x00006dec (928) [4] : 52							
3. 0x0000718c (296) [1] : 34							
4. 0x000072b4 (412) [1] : 36							
5. 0x00007450 (424) [1] : 40							
6. 0x000075f8 (524) [1] : 42							
7. 0x00007804 (592) [1] : 46							
8. 0x00007a54 (584) [1] : 48							
9. 0x00007c9c (285) [1] : 50							
15. 0x00111800 (0x189bf) #A: 7 #F: 2406 (58880, 256, 54784) : 115 : 80							
0. 0x00001d5c (20088) [15] : LARA							
1. 0x00006bd4 (24588) [15] : 37							
2. 0x0000cbe0 (16352) [15] : 37							
3. 0x00010bc0 (14860) [23] : 56							
4. 0x000145cc (16520) [20] : 58							
5. 0x00018654 (496) [3] : 420							
6. 0x00018844 (379) [3] : 422							
16. 0x0000eb800 (0x25e2b) #A: 4 #F: 4000 (59904, 0, 58880) : 131 : 56							
0. 0x00002984 (59640) [15] : LARA							
1. 0x0001127c (54220) [15] : 45							
2. 0x0001e648 (28352) [15] : 419							
3. 0x00025508 (2339) [1] : 433							
17. 0x000db800 (0xfcd) #A: 3 #F: 2097 (59904, 0, 58880) : 108 : 48							
0. 0x000022d8 (16748) [15] : LARA							
1. 0x00006444 (16540) [15] : 45							
2. 0x0000a4e0 (22557) [15] : 419							
18. 0x000c4800 (0x16f63) #A: 6 #F: 2281 (53760, 1024, 42496) : 114 : 72							
0. 0x00002a80 (23052) [15] : LARA							
1. 0x0000848c (20652) [15] : 37							
2. 0x0000d538 (21680) [15] : 37							
3. 0x000129e8 (17308) [23] : 56							
4. 0x00016d84 (272) [1] : 32							
5. 0x00016e94 (207) [1] : 34							

19. 0x000ab000 (0x1952a) #A: 2 #F: 2985 (38400, 5888, 50688) : 119 : 40
1. 0x000030424 (48292) [15] : LARA
1. 0x0000f0c8 (42082) [28] : 54
20. 0x00006a00 (0x4cd9) #A: 4 #F: 540 (11776, -3328, 58880) : 50 : 56
1. 0x000003094 (17664) [15] : LARA
1. 0x000004894 (760) [3] : 90
2. 0x000004b8c (188) [2] : 88
3. 0x000004c48 (145) [1] : 86
21. 0x00008a800 (0x1b084) #A: 2 #F: 4000 (64000, 8960, 62976) : 132 : 40
1. 0x000002968 (25620) [15] : LARA
1. 0x000008d7c (74504) [22] : 84
22. 0x000077000 (0x1307f) #A: 2 #F: 1400 (3584, -3328, 34304) : 96 : 40
1. 0x000002090 (45424) [15] : LARA
1. 0x00000d200 (24191) [15] : 439
23. 0x000073800 (0x37f4) #A: 2 #F: 805 (32256, -24064, 51712) : 58 : 40
1. 0x00000078c (12136) [15] : LARA
1. 0x0000037a8 (76) [1] : 46
24. 0x000067000 (0xc262) #A: 5 #F: 1184 (34304, -1024, 52856) : 92 : 64
1. 0x0000010b0 (21824) [15] : LARA
1. 0x0000063d4 (21176) [15] : 69
2. 0x00000b68c (1848) [1] : 48
3. 0x00000b14c (852) [1] : 46
4. 0x000001c18 (330) [4] : 50
25. 0x00004b800 (0x1b566) #A: 7 #F: 2180 (18944, -1024, 44544) : 113 : 80
1. 0x0000024b0 (13228) [15] : LARA
1. 0x00000585c (11792) [15] : 439
2. 0x00000866c (14248) [15] : 439
3. 0x00000be14 (35244) [23] : 427
4. 0x0000147c0 (27088) [20] : 433
5. 0x00001b190 (344) [3] : 429
6. 0x00001b2e8 (638) [1] : 431
26. 0x000042000 (0x944e) #A: 6 #F: 1770 (49664, 0, 67072) : 107 : 72
1. 0x00ffff (4294960383) [15] : 65535
1. 0x000001b00 (11704) [18] : 426
2. 0x000004a8b (10756) [15] : 416
3. 0x0000072bc (6828) [19] : 422
4. 0x000008688 (1272) [7] : 436
5. 0x000009260 (494) [4] : 420
27. 0x000348000 (0x0d59) #A: 6 #F: 1770 (58880, -4096, 35680) : 105 : 72
1. 0x000001d0c (30584) [15] : LARA
1. 0x000009484 (10464) [15] : 422
2. 0x00000b0d4 (568) [5] : 424
3. 0x00000b19c (208) [1] : 446
4. 0x00000c0b6 (3404) [9] : 419
5. 0x00000cd8b (673) [3] : 432
28. 0x00013800 (0x68eb) #A: 2 #F: 956 (86528, 0, 9728) : 82 : 40


```

1. 0x00004050 (10395) [ 20]: 38
29. 0x0001a800 (0x725b) #A: 2 #F: 825 (89600, 0, 19968) : 76 : 40
   0. 0x00000a54 (16112) [ 15]: LARA
   1. 0x00004944 (10519) [ 15]: 34
30. 0x00022000 (0xa0fa) #A: 3 #F: 718 (79360, 0, 32256) : 73 : 48
   0. 0x00000b2c (21028) [ 15]: LARA
   1. 0x00005d50 (13636) [ 15]: 34
   2. 0x00009294 (3686) [ 15]: 39
31. 0x0002c800 (0x7ee1) #A: 3 #F: 638 (83456, -3584, 72192) : 69 : 48
   0. 0x00007114 (16108) [ 15]: LARA
   1. 0x00004600 (14296) [ 15]: 34
   2. 0x00007dd8 ( 265) [ 1]: 32
32. 0x001b6000 (0x220d3) #A: 9 #F: 3206 (52736, 4864, 79360) : 120 : 96
   0. 0x000036d4 (49256) [ 15]: LARA
   1. 0x000073c (24512) [ 20]: 58
   2. 0x000156fc (1692) [ 10]: 42
   3. 0x00015d98 (4408) [ 7]: 60
   4. 0x00016ed0 ( 228) [ 1]: 54
   5. 0x00016fb4 ( 280) [ 1]: 48
   6. 0x000170cc (10024) [ 15]: 49
   7. 0x00019774 (28876) [ 15]: 51
   8. 0x000208c0 (6163) [ 1]: 62

```

```

33. 0x001d8800 (0xd84f) #A: 2 #F:1500 (44544, 9216, 35328) : 101 : 40
    0. 0x000019c4 (24340) [ 15] : LARA
    1. 0x000078d8 (24439) [ 28] : 54
34. 0x001e6800 (0x185e) #A: 2 #F: 170 (45568, 7680, 23040) : 5 : 40
    0. 0x00000178 ( 5328) [ 15] : LARA
    1. 0x00001648 ( 534) [ 1] : 243
35. 0x001e8800 (0xa890) #A: 2 #F:1418 (78336, 5376, 37376) : 97 : 40
    0. 0x000010e0 (14840) [ 15] : LARA
    1. 0x00004ad8 (23992) [ 28] : 54
36. 0x001f3800 (0x1a18a) #A: 3 #F:1437 (78336, 5376, 51712) : 99 : 48
    0. 0x00001ecc (42280) [ 15] : LARA
    1. 0x0000c3f4 (28848) [ 20] : 48
    2. 0x000134a4 (27878) [ 16] : 46
37. 0x0020e000 (0x2ffcf) #A: 5 #F:4839 (46592, 10752, 26112) : 134 : 64
    0. 0x000042dc (17304) [ 15] : LARA
    1. 0x00008674 (67900) [ 16] : 34
    2. 0x00018fb0 (22832) [ 20] : 36
    3. 0x0001e8e0 ( 1408) [ 5] : 32
    4. 0x0001ee60 (69999) [ 28] : 54
38. 0x0023e000 (0x32651) #A: 4 #F:2156 (78336, 5376, 51712) : 109 : 56
    0. 0x000032f8 (58812) [ 15] : LARA
    1. 0x000118b4 (43532) [ 16] : 46
    2. 0x0001c2c0 (41908) [ 20] : 48
    3. 0x00026674 (49117) [ 28] : 54
39. 0x00270800 (0x2a959) #A: 9 #F:1781 (39424, 15360, 48640) : 106 : 96
    0. 0x00002eec (24812) [ 15] : LARA
    1. 0x00008fd8 (12204) [ 11] : 89
    2. 0x0000bf84 (13712) [ 11] : 89
    3. 0x0000f514 ( 1448) [ 3] : 422
    4. 0x0000ffac (66800) [ 29] : 79
    5. 0x0001ffac (12760) [ 11] : 428
    6. 0x00023184 (15576) [ 11] : 428
    7. 0x00026e5c ( 4092) [ 1] : 173
    8. 0x00027e58 (11009) [ 14] : 424
40. 0x0029b800 (0xb910) #A: 8 #F:1562 (31232, 19200, 69120) : 103 : 88
    0. 0xffffffff (-4294960599) [ 15] : 65535
    1. 0x00001a28 (30928) [ 17] : 44
    2. 0x000092f8 ( 2220) [ 1] : 32
    3. 0x00009ba4 ( 1176) [ 1] : 34
    4. 0x0000a03c ( 1408) [ 1] : 36
    5. 0x0000a5bc ( 3276) [ 1] : 38
    6. 0x0000b288 ( 1280) [ 1] : 40
    7. 0x0000b788 ( 392) [ 2] : 42
41. 0x002a7800 (0x20e24) #A: 2 #F:4141 (29184, 9216, 32256) : 133 : 40
    0. 0x00004954 (51704) [ 15] : LARA
    1. 0x0001134c (64216) [ 15] : 47
42. 0x002c8800 (0x7999) #A: 2 #F:1000 (42496, 19712, 32256) : 84 : 40
    0. 0x00001aa0 ( 1900) [ 15] : LARA
    1. 0x0000220c (22413) [ 25] : 32
43. 0x002d0800 (0xaaec) #A: 2 #F: 552 (36352, 15360, 54784) : 64 : 40
    0. 0x00000b14 (17832) [ 15] : LARA
    1. 0x000050bc (23088) [ 29] : 79
44. 0x002db800 (0x49ed2) #A: 6 #F:6000 (46592, 10752, 26112) : 135 : 72
    0. 0x00006a44 (56660) [ 15] : LARA
    1. 0x00014798 (86276) [ 16] : 34
    2. 0x0002989c (15892) [ 20] : 36
    3. 0x0002d6b0 ( 1460) [ 5] : 32
    4. 0x0002dc64 (110536) [ 28] : 54
    5. 0x00048c2c ( 4774) [ 2] : 426

```