

Adventure Construction Set File Format (Draft)

By Dan Boris (danlb_2000@yahoo.com)

Version 0.5, 11/25/2016

NOTE: This is a early draft version of the file format. It is incomplete and may have inaccuracies. For the most accurate details of the file format please see the source code for my ACSView program at:

<https://github.com/danlb2000/ACSViewer>

1. Introduction

The purpose of this file is to document the Adventure Construction Set game file formats for MSDOS and the C64.

2. Document Conventions

All numbers in this file prefixed with "0x" are hexadecimal, if the number does not have a prefix then it is decimal.

3. Platform Differences

The C64 and MSDOS file formats are almost identical with two exceptions. The .D64 files have an additional 512 (0x200) bytes at the start of the file. All addresses in this document are for the MSDOS version, so you need to add 0x200 to all base addresses to get the location for a C64 file.

The second difference is in the encoding of the graphics. See section 5 for details.

4. Text Encoding

4.1 Unpacked ASCII

Unpacked ASCII data is stored one character per byte. If the byte value is less than 32 then the ASCII value is byte + 64, otherwise the byte contains the ASCII value.

4.2 Packed

Packed text has three character for every two bytes (one word). The word value equals $(C1 * 1600) + (C2 * 40) + C3$, where C1, C2, and C3 are the character values from the following table:

0	{Space}	10	J	20	T	30	
1	A	11	K	21	U	31	
2	B	12	L	22	V	32	
3	C	13	M	23	W	33	
4	D	14	N	24	X	34	
5	E	15	O	25	Y	35	
6	F	16	P	26	Z	36	
7	G	17	Q	27		37	
8	H	18	R	28		38	
9	I	19	S	29		39	

5. General

5.1 3.1 General text

0x16A00 – 0x16AFF: Introduction, unpacked ASCII

0x1C200 – 0x1C211: Adventure name, unpacked ASCII

0x1C215 – 0x1C228: By line, unpacked ASCII

6. Graphics

6.1 Palette

Background: 0x2A802

Color 1: 0x2A803

Color 2: 0x2A804

Color 3: 0x2A805

Color Number	Red	Green	Blue
0	0	0	0
1	0	0	170
2	0	170	0
3	0	170	170
4	170	0	0
5	170	0	170

6	170	85	0
7	170	170	170
8	85	85	85
9	85	85	255
0xA	85	255	85
0xB	85	255	255
0xC	255	85	85
0xD	255	85	255
0xE	255	255	85
0xF	255	255	255

6.2 Pictures PC

A PC ACS game contains 96 pictures. Each picture is 16x16 pixels, and each pixel is one of four colors specified by the palette configuration (See 4.1). Each picture uses 64 bytes and 2 bits per pixel. The based address of each picture = $0x28720 + \text{picture number} * 16$. The pixels are encoded as follows

Lines 0 – 7, Offset 0: pixels 12-15, offset 0x2400: pixels 8-11, offset 8: pixels 4-7, offset 0x2408: pixels 0-3

Lines 8-15: Offset 0x800: pixels 12-15, offset 0x2C00: pixels 8-11, offset 0x808: pixels 4-7, offset 0x2C08: pixels 0-3

6.3 Pictures C64

A C64 ACS game contains 110 pictures. Each picture is 8x16 pixels, and each pixel is one of four colors specified by the palette configuration (See 4.1). Each picture uses 32 bytes and 2 bits per pixel. The based address of each picture = $0x28920 + \text{picture number} * 16$. The pixels are encoded as follows

7. Things

7.1 Thing Names

0x1B800 – 0x1BCFF, 10 bytes per name packed.

7.2 Thing Type

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$

Byte 1:

Bits 0..3: Thing type

0 = Room Floor

1 = Treasure

2 = Magic Item

3 = Missile Weapon

4 = Melee Weapon

5 = Armor

6 = Magic Spell

7 = Portal

8 = Space

9 = Custom Space

10 = Obstacle

11 = Custom Obstacle

12 = Store

15 = Undefined thing

Things 1-32 , person can have 3

Things 33-80 , person can have 1

Things 81 - Many

2.3 Room Floor

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$

Byte 4, bits 0..5: Picture

Messages: None

2.4 Treasure

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$ (PC)

Byte 1:

Bit 4:

- 0 = Doesn't disappear after it's dropped
- 1 = Disappears after it's dropped

Bits 5..6: Weight Multiplier

- 0 = x1
- 1 = x20
- 2 = x-1
- 3 = x-20

Bit 7: Value Multiplier

- 0 = x1
- 1 = x20

Byte 2: Weight base value

Byte 3: Value base value

Byte 4, bits 0..5: Picture

Messages: None

2.5 Missile Weapon/Melee Weapon/Armor

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$

Byte 1:

Bit 4:

- 0 = Doesn't disappear after user
- 1 = Disappears after use

Bits 5..6: Weight Multiplier

- 0 = x1
- 1 = x-1
- 2 = x20
- 3 = x-20

Bit 7: Value Multiplier

0 = x1

1 = x20

Byte 2: Weight base value

Byte 3: Value base value

Byte 4: Index of extended properties

Address2 = 0x1C084 + Index * 3

Byte 1:

Bits 0..3 = Power (Missile Weapons)

Bits 0..4 = Power (All others)

Bits 5..7 = Range - 2 (*Missile Weapon Only*)

Byte 2:

Bits 0..3: Chance of breaking

0xF = 15%

0x0 = 0%

Bits 4..7: Attack Adjustment (Weapon)

0 = -35%

1=-30

2=-25

3=-20

4=-15

5=-10

6 = -5%

7 = 0

8 = 5%

F = 40%

Bits 4..7: Attack Adjustment (Armor)

0 = -65%

1=-60%

2=-55%

3=-50%

4=-45%
5=-30%
6 = -25%
F = 10%

Byte 3:

Bits 0..5 = Picture #

Bit 6: (0 = Only Usable by it's owner/1 = Used by anyone)

Bit 7: (0 = Not Magic/1 = Magic)

Message: None

Magic Item

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$

Byte 1:

Bit 4:

0 = Doesn't disappear after user

1 = Disappears after use

Bits 5..6: Weight Multiplier

0 = x1

1 = x20

2 = x-1

3 = x-20

Bit 7: Value Multiplier

0 = x1

1 = x20

Byte 2: Weight base value

Byte 3: Value base value

Byte 4: Index to extended properties

Offset = value at 0x1C082 (possible two byte at 82 and 83)

Address2 = 0x1C084 + Offset + (Index * 3)

Byte 1, bits 0..3: Spell Action

Bit 7: 1 = Invoked when owner uses the item

Byte 2:

0..5: Picture

6: 1 = Invoked when item is dropped

7: 1 = Invoked when item is picked up

Byte 3: Spell Action Parameter

Message 0: Not used

Message 1: Show when spell is cast

Magic Spell

Address = 0x1BD00 + (Thing Number - 1) * 4

Byte 1:

Bit 4:

0 = Doesn't disappear after user

1 = Disappears after use

Byte 2: Power to cast

Byte 3: Action (See Spell Actions)

Byte 4: Action Parameter (See Spell Actions)

Message 0: Not used

Message 1: Show when spell is cast

Portal

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$

Byte 1:

Bit 5: 0 = One Way/1 = Two Way

Byte 2:

Bit 0..5 = Picture

Bit 6..7 = Open To

0 = Anyone may pass, no spell invoked

1 = Invoke spell when someone moves here

Byte 3: Action (See Spell Actions)

Byte 4: Action param (See Spell Actions)

2 = Open to owners of a specific item

Byte 4: Thing number

3 = Open to those who don't own a . . .

Byte 4: Thing number

Byte 3:

Bit 7: Destroy thing

Address2 = $0x1BF00 + (\text{Thing Number} - 1)$

Bit 7: 1 = Don't tell why can't pass

0 = Tell why can't pass

Message 0: Why passage is blocked

Message 1: Not used

Space

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$

Byte 2:

Bit 0..5 = Picture

Bit 6..7 = Open To (See Portal)

Address2 = $0x1BF00 + (\text{Thing Number} - 1)$

Bit 7: 0 = Don't tell why can't pass

1 = Tell why can't pass

Message 0: Not used

Message 1: Show when spell is cast

Custom Space

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$

Byte 0:

Bit 7: 0 = Choose object when thing put in room

1 = Choose spell modifier when thing used

Byte 1:

Bit 0..5 = Picture

Bit 6..7 = Invoke when

0 = Invoke spell when thing dropped here

1 = Invoke Spell When Someone Moves Here

2 = Open to owners of a specific thing

3 = Open to those who don't own a

Byte 2:

Bit 0..3: Spell Action

Bit 7: 1 = Destroy thing needed to move here

Byte 3: Open To Parameter/Spell Action Parameter

Address2 = $0x1BF00 + (\text{Thing Number} - 1)$

Bit 0..6: Message Index (Message won't be displayed = 0)

Bit 7: 1 = Don't tell why can't pass
0 = Tell why can't pass

Message 0: Reveal why passage blocked
Message 1: Show when spell is cast

Obstacle

Address = $0x1BD00 + (\text{Thing Number} - 1) * 4$

Byte 2:

bits 0..5: Picture

bit 6: 1 = Bumping into obstacle invokes spell

Byte 3:

bits 0..3: Spell Action

Byte 4: Spell Action Parameter

Message 0: Reveal why passage blocked
Message 1: Show when spell is cast

Custom Obstacle

Message 0: Not Used
Message 1: Show when spell is cast

Store

Messages: Not used

Message

Messages are used by Magic Items, Magic Spells, Portal, Space, Custom Space, Obstacle, Custom Obstacle

address = $0x1BF80 + (\text{Thing Number} - 1)$

bit 0..6: Message Index

bit 7: 1 = Show Message/0 = Don't Show Message

(Message index might be -1)

Message Address = $0x16B00 + (((\text{Message Index} \& 0x7E) \gg 1) * 0x100) + (\text{Message Index} \& 0x01) * 0x78$

0x16B00 – 0x192FF

Spell Actions

Spell actions are used by Magic Items, Magic Spells, Portals, Spaces, Custom Spaces, Obstacles

Action

0 = Do Nothing

1 = Kill anyone but an owner of

Parameter = Item Number

2 = Summon or Banish Creature

Parameter = Creature (0 = All)

3 = Increase Magical Defense, but not above

Parameter:

0 = No Magic Defense

1 = Non-Magical Weapons Do Half Damage

2 = Only Magic Weapons Do Damage

3 = Magic Weapons Do Half Damage

4 = Decrease Magical Defense, but not below

Parameter:

0 = No Magic Defense

1 = Non-Magical Weapons Do Half Damage

2 = Only Magic Weapons Do Damage

3 = Magic Weapons Do Half Damage

5 = Increase Victim's [stat]

Parameter:

0 = Constitution

1 = Strength

2 = Dexterity

3 = Speed

4 = Wisdom

5 = Dodge Skill

6 = Parry Skill

7 = Armor Skill

8 = Melee Skill

9 = Missile Skill

6 = Decrease Victim's [stat]

Parameter:

0 = Constitution

1 = Strength

2 = Dexterity

3 = Speed

4 = Wisdom

5 = Dodge Skill

6 = Parry Skill

7 = Armor Skill

8 = Melee Skill

9 = Missile Skill

7 = Change Power of victim by

Parameter

0..5: Amount

6: 0 = Permanently/1 = Temporarily

7: 0 = Positive/1 = Negative

8 = Change life force of victim by

Parameter:

0..5: Amount

6: 0 = Permanently/1 = Temporarily

7: 0 = Positive/1 = Negative

9 = Give To Victim one

Parameter: Item to give

A = Display long message

Parameter: Message number

B = Play music

C = Rid room of every uncarried

D = Add to room one

E = Activate all things at this place

ACS Adds

0x1C000 – 0x 1C07F

Address = 0x1C000 + (Thing Number -1)

Bits 4..7: Numbers of a thing that ACS can add to a game. 0xF = Add Many.

Long Messages

0x6c00 – 256 Ascii characters for each message

8. World Map

8.1 Terrain

0x1C700 – 0x1C71F – 16 terrain types, 2 byte each

Byte 0 - 0..4: Picture number

- 6..7: Type

0 = Impassible

1 = Open to travel by all

2 = Open only to owners of (object number in byte 1)

3 = Triggers the spell (object number in byte 1)

Byte 1: Terrain Type Parameter

8.2 Map

The world is 40 tiles by 40 tiles.

6.3 Map portals (32 max)

1C720-1C75F – Portal x,y positions (FF = not used)

1C720 – Portal 1 X position

1C721 – Portal 1 Y position

1C722 – Portal 2 X position

1C723 – Portal 2 Y position

1C760-1C79F – Portal destination positions (FF = not used)

1C760 – Portal 1 destination X position

1C761 – Portal 1 destination Y position

1C762 – Portal 2 destination X position

1C763 – Portal 2 destination Y position

1C7AA-1C7E9 – Portal Destination region/room

1C7AA- 0...3 – Destination room in region

4...7 – Destination region

1C7AB – 0...3 – X position in destination room

4...7 – Y position in destination room

1C

5.0 Regions

0x1C229 = Number of regions (Max 15)

5.1 Names

Address = 0x1C441 + 20 * region number

20 character name in unpacked ASCII

5.2 Rooms

Region Address = 0x1D300 + 0xC00 * region number

Byte 0: Number of rooms in region (Max 16)

Byte 1,2: Add to Region Address to get the first byte after the end of the room item definitions

5.2.1 Room name

Name Address = Region Address + 0x3F3 + 10 * room number

15 character room name in packed ASCII

5.2.2 Room properties

Maximum 16 rooms per region

Room Address = Region Address + 9 + (room number * 5)

Byte 0: Wall picture

Byte 1: bits 0..3 = Room Height (0 if deleted)

bits 4..7 = Room Width (0 if deleted)

Byte 2: X Position of Room (0 if deleted)

Byte 3: Y Position of Room (0 if deleted)

Byte 4: Random Creature 0..3 - Chance of appearing (value * 5 = Chance)

4..6 – Creature number in region

5.2.3 Room Resident Creatures

16 resident creatures per region

Region Address = $0x1D300 + 0xC00 * \text{region number}$

Region Address + 7: Resident Creature Pointer

Creature Name address = $\text{Region Address} + 0x453 + \text{pointer} + (\text{Creature Number} * 10)$

Creature Data Address = $\text{Region Address} + 0x1A3 + (\text{Creature Number} * 37)$

5.2.4 Room Items

Region

Room Items Record:

Byte 0,1: Number of bytes used by the rooms items including these two bytes

Item Record:

Byte 0: bits 0..3: Y position

Bits 4..7: X position

(If 0xFF then item was deleted)

Byte 1: bits 0..6: Item Number

Byte 2: Item parameter

6.0 Creatures

6.1 Class Names

0x19300 – 0x19378, 15 bytes per name, unpacked ASCII.

6.1 Name

0x1B280 – 0x1B77F

15 characters packed into 10 bytes

6.2 Class

0x1B200 – 0x1B27F – Class number for each creature. 8 = Creature entry not used.

6.3 Properties

Address = $0x19400 + ((\text{Monster Number} / 6) * 0x100) + ((\text{Monster Number} \% 6) * 37)$

0x19400 – 0x1A67F, 37 bytes per creature

Offset

0x00

0..5: Constitution

6..7: Special Defense

0 = No magical defense

1 = Non-Magic weapons do half damage

2 = Only magic weapons do damage

3 = Magic weapons do half damage

0x01

0..4: Strength

0x02

0..4: Dexterity

0x03

0..5: Life Force

0x04

0..3: Speed

4..7: Strategy

Byte 4 (7):

0 = Cautious

1 = Brave

Byte 4 (6):

0 = Peaceful

1 = Aggressive

Byte 4 (4..5):

0 = Thief

1 = Neutral

2 = Enemy

3 = Friend

0x05

0..6: Power

0x06

0..6: Dodge Skill

0x07

0..6: Parry Skill

0x08

0..6: Armor Skill

0x09

0..6: Melee Skill
7: Mimic Power

0x0A

0..6: Missile Skill

0x0B: ?

0x0C: ?

0x0D: position X

0x0E: position Y

0x0F:

0..4: Size

0x10: 0..3 – Room Number?, 4..7 – Region?

0x11

0..4: Wisdom

0x12-0x13: Wealth

0x14: Weapon

0x15: Armor

0x16 - Picture

Monster Class Names

8 class names with 15 characters per name. Names are ASCII encoded starting at 19300.

Notes

Regions

Maximum 16 creatures per region

Maximum 8 random creatures per region

Thing Name

Thing 1 Name: 1B800-1B809

Thing 2 Name: 1B80A-1B813

Thing 127 Name: 1BCEC-1BCF3

Thing Name: 15 characters stored in 10 bytes (64 bits)

Characters allowed: A-Z, 0-9, ., -

