

The Blades of Avernum Data Format

Arancaytar

September 25, 2013

Abstract

The following is a detailed documentation of the data format used to store Blades of Avernum scenario files, known as BAS files. This document should suffice to allow anyone to create a program that reads or writes a BAS file for analysis, manipulation or representation.

Part I

Introduction

The BAS file is a concatenation of data objects with fixed length. There are three such chunk types:

Global: There is exactly one global chunk, and it occupies the first 7480 bytes of the file.

Outdoor: There are up to 100 of these.¹ They are stored sequentially, and each is 12892 bytes long.

Town: There are up to 200 of these.² They are stored sequentially. Each consists of one chunk of 16346 bytes, followed by one chunk of either 20480, 11520 or 5120 bytes. The town sizes - required to find the offsets of these chunks - are stored in the Global chunk.

1 Data Types

Unless otherwise specified, all data types in this format are integers composed of 1, 2 or 4 bytes. The remainder are usually fixed-length ASCII (0x00-0x7f) strings.

Strings are null-terminated. Editors may not bother to zero out a string while overwriting it, so make sure that strings are properly truncated at the first null byte.

¹The data format can theoretically specify an outdoor map size up to 32767x32767. The official specification limits the number to 100. The game engine has been found to accept more, but in the interest of compatibility, applications should never attempt to write files with more than 100 sections.

²As we shall see, this limit is hard-coded in the format and cannot be expanded - even by a game engine patch - without breaking compatibility.

Part II

Global Data

This chunk always occupies the first 7480 bytes of the file; the chunk offsets are therefore file offsets.

2 0x0000..0x000a: Version Information

The first ten bytes of the file encode version information that is not particularly useful. As we see, all of the below is either fixed to a specific value or meaningless.

0x0000..0x0003: Platform

The first four bytes of the chunk serve as a platform identifier. The source comments of the editor state that this sequence is supposed to be 0x61d70721 for Mac, and 0xc73d0235 for Windows; the difference lies in endianness. In practice, the Windows form is obsolete. BAS files are universally written in Mac form (with big-endian shorts), and all files begin with the former sequence.

0x0004..0x0006: Scenario Version

These three bytes identify the scenario version. This is interpreted and printed as a three-sequence string, where each sequence should not exceed 9. This field has no significance and is only used for display.

0x0007: Engine Version

This byte identifies the minimum required version of the game engine. The correspondence between BoA game version and engine version is not documented. The current and likely last commercial versions of the BoA game are 1.2.1 for Mac and 1.0.1 for Windows. This field should be set to 1. Behavior for values other than 1 is undocumented.

0x0008..0x000a: Editor Version

These three bytes identify the version of the editor that last wrote the file. As multiple editors with independent version histories can edit files, this data is currently meaningless for most purposes.

3 0x000b..0x000f: Size

0x000b: Number of Towns

This byte simply specifies how many towns the scenario contains. The value cannot be greater than 200.

0x000c..0x000d: Outdoor Width

Horizontal (west-east) dimension of the outdoor map. As all short integers in this format, it is signed and big-endian. This value should not be greater than 100.

0x000e..0x000f: Outdoor Height

Vertical (north-south) dimension of the outdoor map.

4 0x0010..0x0015: Rating

All these are short integers for unknown reasons, because they'd easily fit inside a byte.

0x0010..0x0011: Minimum Level

The minimum recommended level for players.

0x0012..0x0013: Maximum Level

The maximum recommended level for players.

0x0014..0x0015: Content Rating

The scenario's declared content rating. The following values are valid:

- 0: Everyone
- 1: Teen
- 2: Mature
- 3: Adult

5 0x0016..0x174f: Text

This contains the scenario name, description, credits and other things. Nestled between these is the scenario icon.

0x0016..0x0047: Scenario Name

50 characters for the scenario name.

0x0048..0x0049: Scenario Icon

The scenario icon number. This value chooses one from a fixed list of icons and cannot be customized further. The value corresponds to the number of the graphics file.

0x004a..0x0149: Description

256 characters for the scenario description.

0x014a..0x0349: Credits

Two strings of 256 characters for the credits. These strings will be displayed as paragraphs, so they should not be treated as a single 512-character string.

0x034a..0x0549: Comments

Two strings of 256 characters for the comments.

0x054a..0x054f: Introduction Screens

Three 2-byte integers that encode the custom graphics displayed during the intro sequence.

0x0550..0x174f: Introduction Texts

Three times six 256 character strings displayed during the introduction sequence. Each screen will show six paragraphs of text.

6 0x1750..0x18df: Town Data

0x1750..0x1817: Town Sizes

These 200 bytes store the size of each town, and are responsible for the hard-coded 200 town limit. Valid values are:

- 0: Large, 64x64
- 1: Medium, 48x48
- 2: Small, 32x32

These values determine the way in which the remaining file data is sectioned, and incorrect values at this point may corrupt the file beyond easy recovery.

0x1818..0x18df: Town Hidden Flags

These 200 bytes determine whether each town starts out as hidden. Town entrances associated with hidden towns are invisible and inaccessible until the town is made visible by a script call. Valid values are:

- 0: Visible
- 1: Hidden

7 0x18e0..0x18e7: Spawn Location

0x18e0..0x18e1: Starting Town

The number of the town the player begins the game in. (This is a two-byte value even though the 200 town limit is the most hard-coded limit in the format.)

0x18e2..0x18e3: Indoor Coordinates

These two bytes are an (x,y) coordinate pair determining the spot inside the town that the player begins on.

0x18e4..0x18e5: Outdoor Section

The outdoor section (coordinate pair) that the player will be placed in once they leave town.

0x18e6..0x18e7: Outdoor Coordinates

The coordinates inside the outdoor section that the player will be placed in.

8 0x18e8..0x1923: Variable Town Entries

There are ten slots for variable town entries. Each can modify a specific town with an SDF increment, such that the player enters the incremented town instead of the original one. To avoid unintended behavior, unused slots should be set to an invalid town number such as 255.

0x18e8..0x18fb: Modified Town Number

The town modified by each slot. These are two-byte integers. The effects of using the same town in more than one slot are undocumented.

0x18fc..0x1923: Modifying SDF

The flag used to increment the town number in each slot. Each flag is defined by two short integers (thus four bytes) ranging from 0 to 300 and 0 to 30, respectively (refer to the BoA documentation).

9 0x1924..0x1b7b: Horses and Boats

These 600 bytes store the status of the horses and boats. Each of the 30 horses and 30 boats has one record, which will be explored in detail. Note that while the format allows these entities to exist outdoors or in town (or both), the documentation specifies that they must initially be in a town.

0x1924..0x1a4f: Horses #1 to #30

Each record consists of 10 bytes.

00..01 The town coordinates of the horse.

02..03 The outdoor coordinates.

03..05 The outdoor section coordinates.

06..07 The town number.

08..08 Whether the horse exists (boolean).

09..09 Whether the horse belongs to the player (boolean).

0x1a50..0x1b7b: Boats #1 to #30

The boat records follow the same format.

10 0x1b7c..0x1d37: Editor Information

0x1b7c..0x1d33: Item Placement Shortcuts #1 to #10

This defines shortcuts inside the official scenario editor to place items. Alternative editors may choose to implement the same functionality, but otherwise this data can be safely ignored.

There are ten such shortcuts, each 44 bytes long. Each shortcut is able to place up to ten item types.

00..01 The terrain type the shortcut applies to (short).

02..15 The item types that may be placed (10 shorts).

16..29 The probabilities (in percent, between 0 and 100) of placing that item (10 shorts).

2a..2b Whether the items are generated as belonging to the player (short, but actually boolean).

0x1d34..0x1d35: Last Outdoor Section

The coordinates of the outdoor section that was last edited. Editors may choose to load this section when initializing, for convenience.

0x1d36..0x1d37: Last Town

The last town that was edited.

Part III

Outdoor Section

The next part of the file consists of up to 100 outdoor sections, each taking 12892 bytes. They are sequentially stored by rows, from west to east and north to south. Offsets are specified from the start of the section, which is `0x1d38 + 0x325cn` for section $n \in \mathbb{N}$.

0x0000..0x0013: Name

A 20 character string for the name of the outdoor section.

11 0x0014..0x2413: Ground Data

0x0014..0x0913: Floor Grid

A 48x48 grid of bytes for the floor cells. They are sequentially stored by *columns* (to be addressable as `[x][y]`), from north to south and west to east.

0x0914..0x1213: Height Grid

A 48x48 grid of bytes for the height levels.

0x1214..0x2413: Terrain Grid

A 48x48 grid of shorts for the terrains. These are layered on top of the floors.

12 0x2414..0x2d9f: Special Areas

0x2414..0x2503: Special Rectangles #1 to #30

Thirty special encounter areas defined by their boundaries, inclusive. Each takes 8 bytes.

0..1 Top (north) bound.

2..3 Left (west) bound.

4..5 Bottom (south) bound.

6..7 Right (east) bound.

0x2504..0x2539: Special States #1 to #30

Thirty special encounter states (shorts) triggered by walking over the corresponding rectangle.

0x2540..0x257f: Town Entrance Rectangles #1 to #8

Eight rectangles that cause players to enter a town when walked into. Each is stored as above.

0x2580..0x258f: Town Entrance Destinations #1 to #8

The towns (shorts) that the player will enter when walking over the entrances.

0x2590..0x259f: Sign Locations #1 to #8

The coordinates of each sign. These will usually be initialized to 0x00ff.

0x25a0..0x2d9f: Sign Texts #1 to #8

The text of each sign.

13 0x2da0..0x30ff: Outdoor Encounters

There are 16 outdoor encounters per section: 4 wandering (recurring) encounters, 4 special encounters and 8 presets, in that order. Each of these is stored in the same record type, which is 54 bytes long and will be explored in detail.

0x2da0..0x2e77: Wandering #1 to #4

0x2e78..0x2f4f: Special #1 to #4

0x2f50..0x30ff: Preset #1 to #8

00..07: Hostile Types #1 to #4

08..0f: Hostile Amounts #1 to #4

10..15: Friendly Types #1 to #3

16..1b: Friendly Amounts #1 to #3

1c..1d: Whether the player is forced to fight.

1e..21: SDF that disables the encounter.

22..25: SDF that is set by finishing the encounter.

26..27: Whether the encounter starts instantly (instead of waiting for contact).

28..29: Whether the encounter is to be checked every turn instead of every 10 turns.

2a..2b: Move type:

0: Seek

1: Stationary

2: Roam

3: Follow Road

4: Flee Party

10+n: Do n while staying within ten spaces of spawn.

2c..2d: Start Location (This field is ignored for wandering encounters; the spawn points are used instead).

2e..2f: State when encountered

30..31: State when defeated

32..33: State when fled

34..35: Chance (0-100%) of moving randomly instead of its normal move type.

0x3100..0x3107: **Spawn Points #1 to #4**

These 8 bytes represent coordinate pairs for the four spawn points for wandering monsters.

14 0x3108..0x325b: **Other Data**

0x3108..0x3147: **Room Rectangles #1 to #8**

Eight rectangles that will show a special location text while the player is inside them.

0x3148..0x3237: **Room Texts #1 to #8**

Eight 30 character strings that will be shown when the player is in the corresponding area.

0x3238..0x3245: **Outdoor Section Script**

A 14 character string for the name of the script that will be used in this outdoor section.

0x3246..0x3247: **On Surface**

Whether the outdoor section is on the surface. The game engine likely uses this for ambient sounds and lighting.

0x3248..0x3249: **Region Number**

0x3249..0x325b: **Unknown**

The outdoor section concludes with an array of 10 shorts. The first is the region number; the other nine are never explained.