Формат бинарных данных Atom Binary Stream

(версия документа 2020.10.12, версия формата 2)

Из Атома передаётся поток данных. Первые 3 байта данных являются заголовком, они всегда: 'A', 'B', 'S'. «ABS» обозначает Atom Binary Stream. 4-й байт — версия формата. Начиная с 5 байта и далее идут переменные и их значения.

Байт	Значение	Комментарий
1	'A'	Заголовок, всегда 'A'
2	'B'	Заголовок, всегда 'B'
3	'S'	Заголовок, всегда 'S'
4	0x02	Версия, на текущий момент 0x02
5 и далее	(разное)	Переменные, их имена и значения, а также брекеты

Переменные и брекеты

Типы переменных:

Переменные бывают **одиночными** значениями и **массивами**.

Каждая переменная имеет тип, который в данных обозначается первым байтом, он называется идентификатором. Для того, чтобы было легче и нагляднее работать с идентификаторами, сделано две вещи:

- Идентификатор соответствует первой букве распространённого именования типа.
- Для массивов используется прописной символ, а для одиночных значений — строчный.

Для всех значений всех переменных числового типа используется нотация big-endian.

Идентификатор типа в данных	Именование	Размер, в байтах	Комментарий
'b'	byte	1	Байт (0-255)
'i'	int	4	32 битное целое, знаковый.
'l'	long	8	64 битное целое, знаковый
'f'	float	4	Дробное одинарной точности
'd'	double	8	Дробное двойной точности
's'	string	(разный)	Строка в кодировке UTF-8
'B'	array of bytes	N × 1	Массив байт
'I'	array of ints	N × 4	Массив 32 битных знаковых целых
'L'	array of longs	N × 8	Массив 64 битных знаковых целых
'F'	array of floats	N × 4	Массив дробных одинарной точности
'D'	array of doubles	N × 8	Массив дробных двойной точности
'S'	array of strings	(разный)	Массив строк

Формат записи строк

Все строки, включая имена переменных, записываются в формате:

- 4 байта размера UTF-8 строки в байтах в виде целого 32 битного числа, нотация big-endian. Определяет реальный размер данных в байтах.
- данные строки в кодировке UTF-8.

Удобство UTF-8 в том, что для латинских символов размер строки и её длина совпадают, а каждый латинский символ занимает всего 1 байт.

Формат одиночных переменных

Переменные с одиночными значениями имеют следующий формат:

Байт	Значение	Комментарий
1	Идентификатор типа, например, 'i'	Идентификатор, обозначающий тип переменной (см. ниже).
2 … 5 (т.е. 4 байта)	Размер названия переменной (L) в байтах	Размер указывается 32битным целым числом, нотация big-endian.
6 … (6+L)	Название переменной, например 'myvar'	Имя переменной в виде строки в кодировке UTF-8.
(7+L) …	Значение переменной	Размер переменной зависит от типа

Формат переменных-массивов

Переменные с массивами имеют следующий формат:

Байт	Значение	Комментарий
1	Идентификатор типа, например, 'D'	Идентификатор, обозначающий тип переменной (см. ниже).
2 … 5 (т.е. 4 байта)	Размер названия переменной (L) в байтах	Размер указывается 32битным целым числом, нотация big-endian.
6 … (6+L)	Название переменной, например 'myarray'	Имя переменной в виде строки в кодировке UTF-8.
(7+L) … (10+L) (т.е. 4 байта)	Количество элементов массива N	Количество элементов массива, указывается 32битным целым числом, нотация big-endian.
(11+L) …	Элементы массива	Размер элементов зависит от типа

Брекеты

Брекет — это разделитель, который организует вложенность и разбиение на логические блоки, используется в первую очередь для удобства. Брекетов 2 вида:

- Брекет начала (идентификатор '<'). За ним следует его имя в виде строки с форматом записи строк аналогичным другим переменным. Имя обобщает какую-то логическую сущность (спектр, колонка, линия и что угодно).
- Брекет конца (идентификатор '>'). Не содержит дополнительных данных.

Количество брекетов начала и конца должно совпадать в полной передач. Обязанность контроля соответствия лежит на передающей стороне. Максимальная вложенность брекетов не ограничена.

Пример использования брекетов

Предположим, требуется передать 2 колонки с линиями из таблицы анализа, каждая из которых, для простоты, имеет две характеристики: идентификатор столбца (целое) и имя элемента (строка). Тогда можно использовать брекеты, например, следующим образом:

Логическое представление:

```
<columns
    <column
        id = 1
        element = "W"
    >
    <column
        id = 2
        element = "Al"
    >
>
```

Бинарное представление:

Пояснение	Примерный вид бинарных данных
Заголовок и версия	ABS 0x02
<columns	**<** (0x00 0x00 0x00 0x07) columns
<column	**<** (0x00 0x00 0x00 0x06) column
id = 1	**i** (0x00 0x00 0x00 0x02) id (0x00 0x00 0x00 0x01)
element = "W"	**s** (0x00 0x00 0x00 0x07) element (0x00 0x00 0x00 0x01) W
>	**>**
<column	**<** (0x00 0x00 0x00 0x06) column
id = 2	**i** (0x00 0x00 0x00 0x02) id (0x00 0x00 0x00 0x02)
element = "Al"	**s** (0x00 0x00 0x00 0x07) element (0x00 0x00 0x00 0x02) Al
>	**>**
>	**>**

Таки образом, удалось логически обернуть мета-массив из нескольких (в примере — двух) столбцов. Количество элементов не ограничено.

Комментарии

- Все целые типы и дробные типы (float и double) передаются в big endian, т.е. старшие разряды идут раньше. Вообще в спецификации нет мест, где использовался бы little-endian.

Пример данных (версия 1, устарело)

Примеры бинарных файлов можно получить использовав программу отправки примеров и дампа результата. На всякий случай отдельный простой пример с пояснениями.

Предположим передаётся 2 переменных с целыми числами:

- целая знаковая 32 битная переменная «lightness» равная 3
- целая знаковая 32 битная переменная «darkness» равная 5.

Тогда бинарный поток будет следующий:

```
0000000000: 41 42 53 01 69 00 00 00 | 09 6C 69 67 68 74 6E 65  ABS☺i   ○lightne
0000000010: 73 73 00 00 00 03 69 00 | 00 00 08 64 61 72 6B 6E  ss   ♥i   ◘darkn
0000000020: 65 73 73 00 00 00 05    |                         ess   ♣
```

- ABS (это заголовок)
- 0x01 (версия)
- 0x69 = 'i' (тип целого, 32 бита)
- 0x00 0x00 0x00 0x09 (размер имени переменной — девять символов)
- 'lightness' (имя переменной из девяти символов)
- 0x00 0x00 0x00 0x03 (значение переменной = 3)
- 0x69 = 'i' (тип целого, 32 бита)
- 0x00 0x00 0x00 0x08 (размер имени переменной — восемь символов)
- 'darkness' (имя переменной из восьми символов)
- 0x00 0x00 0x00 0x05 (значение переменной = 5)