

# BDIO

Tomasz Korzec<sup>a</sup>, Hubert Simma<sup>b</sup>

(a) Department of Physics, Bergische Universität Wuppertal,  
Gaußstr. 20, 42119 Wuppertal, Germany

`korzec@uni-wuppertal.de`

(b) John von Neumann Institute for Computing (NIC), DESY,  
Platanenallee 6, 15738 Zeuthen, Germany

`hubert.simma@desy.de`

October 17, 2018

## Abstract

BDIO (binary data input/output) is a lightweight file format for binary data. It provides a thin layer of meta-data, which make the files machine independent and browsable. This document describes version 1.0 of BDIO.

## 1 BDIO file format, version 1.0

A BDIO file begins with a header record, which is followed by zero or more header or data records.

### 1.1 Header record

The first 8 bytes of a header record have the structure

byte	bit7							bit0
	v							v
0	[m7	m6	m5	m4	m3	m2	m1	m0 ]
1	[m15	m14	m13	m12	m11	m10	m9	m8 ]
2	[m23	m22	m21	m20	m19	m18	m17	m16]
3	[m31	m30	m29	m28	m27	m26	m25	m24]
4	[17	16	15	14	13	12	11	10 ]
5	[s	s	s	s	l11	l10	l9	l8 ]
6	[v7	v6	v5	v4	v3	v2	v1	v0 ]
7	[v15	v14	v13	v12	v11	v10	v9	v8 ]

with

m0-m31 Magic number, to identify the file as a BDIO file. Must be: 0x7ffbd07e

v0-v15 BDIO version number, Must be 0x0001

10-111 Remaining (starting from byte 8) length of the header in bytes

s Spare bits (undefined)

Optionally (always if created with this library) the header contains further on:

- 4-Byte little endian int: 0 at the moment (will be dir info in future versions)
- 4-Byte little endian int: creation date (unix time)
- 4-Byte little endian int: last modification date (unix time)
- 0-terminated string: creation user
- 0-terminated string: last modification user
- 0-terminated string: creation host
- 0-terminated string: last modification host
- 0-terminated string: user supplied protocol-info
- padding with 0's until end of header

## 1.2 Data record

Each data record begins with four bytes (short records) or eight bytes (long records):

byte	bit7							bit0
	v							v
0	[f3	f2	f1	f0	lr	s	s	mg ]
1	[13	12	11	10	u3	u2	u1	u0 ]
2	[111	110	19	18	17	16	15	14 ]
3	[119	118	117	116	115	114	113	112]

with

mg Magic bit, to distinguish data records from header records, this bit must be 1

s Spare (undefined)

lr Long record flag. 0=short record, 1=long record

f0-f3 Record format. The choices are

0x0 generic binary

0x1 executable

0x2 int32, big endian

0x3 int32, little endian

0x4 int64, big endian

0x5 int64, little endian

0x6 float32, big endian

0x7 float32, little endian

0x8 float64, big endian

0x9 float64, little endian

0xA generic ascii

0xB XML

0xC spare

0xD spare

0xE spare

0xF spare

u0-u3 User supplied "user info" 0-15

10-119 Remaining length of record in bytes (starting with the first data byte).  
If lr=0, this is all.

If lr=1, the following 4 bytes contain the high bits of the record length

byte	bit7	bit0
	v	v
4	[127 126 125 124 123 122 121 120]	
5	[135 134 133 132 131 130 129 128]	
6	[143 142 141 140 139 138 137 136]	
7	[151 150 149 148 147 146 145 144]	

From then on, the actual data content of a data record starts.

## 2 Remarks

- Only the initial header record is of practical relevance. Headers in the middle of the file are ignored. They are allowed by the format specification, so that simple concatenation of two BDIO files yields a valid BDIO file.
- The “protocol info” string gives the user the possibility to describe what is in the file. E.g. “CO2 concentrations. 1 record per year. Each record contains 365 measurements.” or “Generic Correlator Format 1.0”.
- The `uinfo` bits inside the data record give the user the opportunity to distinguish different records of the same type. E.g. `uinfo=0` could be CO2 concentrations and `uinfo=1` the average temperatures, both saved as big-endian double precision floating point numbers.
- The shortest possible valid BDIO file has a length of 8 bytes
- The longest short data record can have 1048575 bytes of data ( $\approx 1$  MB)
- The longest long data record can have 17592186044415 bytes of data ( $\approx 17$  TB)
- It is highly recommended for bdio file names to have the ending `.bdio`