# Overview

The openPDC .D historical file format consists of several uniformly-sized blocks of data, followed by a file allocation table at the tail of the file.

**Note**: Data is written to the file in Little Endian byte order.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Block | Data Block | Data Block | … | Data Block | File Allocation Table  (Block Map) | File Allocation Table  (Footer) |

The individual potions of the openPDC .D historical file are described below, in order from the tail of the file to the start.

# File Allocation Table

The file allocation table describes the structure of the file. It consists of a fixed 32-byte footer that is preceded by a variable length portion that serves as a map of the data blocks in the file.

## File Allocation Table (Footer)

This portion of the file allocation table is contained in the last 32 bytes of every openPDC .D historical file.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|  | **TimeTag** FileStartTime | | | | | | | |
| 8 | **TimeTag** FileEndTime | | | | | | | |
| 16 | **Int32** PointsReceived | | | | **Int32** PointsArchived | | | |
| 32 | **Int32** DataBlockSize | | | | **Int32** DataBlockCount | | | |

* FileStartTime – The timestamp associated with the first point recorded in the file. OR The timestamp associated with the first data block in the file. This time will match the value of the TimeStamp portion of the first entry in the block map.
  + TimeTag (8 bytes) – A double precision floating point number, representing the number of seconds and milliseconds elapsed since midnight on January 1, 1995.
* FileEndTime – The timestamp associated with the last point recorded in the file. OR The timestamp associated with the last data block in the file. This time will match the value of the TimeStamp portion of the last entry in the block map.
  + TimeTag (8 bytes) – A double precision floating point number, representing the number of seconds and milliseconds elapsed since midnight on January 1, 1995.
* PointsReceived – A count of the points received by <some sensor or input adapter> during the time this file was recorded.
* PointsArchived – A count of the points archived in this file.
* DataBlockSize – The size, in kilobytes, of each data block in this file.
  + Generally, data blocks are 8192 bytes (8KB), as an optimization for disk I/O.
  + In this case, the value in the DataBlockSize field is 8.
* DataBlockCount – A count of the data blocks archived in this file.

## File Allocation Table (Block Map)

This portion of the file allocation table immediately precedes the footer section described above. It is variable in length, depending on the number of data blocks in the file.

The block map consists of a 10 byte header, followed by a series of 12 byte block description entries.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Block Map Header (10 bytes) | Data Block Description (12 bytes) | Data Block Description (12 bytes) | Data Block Description (12 bytes) | Data Block Description (12 bytes) | … | Data Block Description (12 bytes) |

### Block Map Header

Header description goes here.

### Data Block Description

Each data block description entry is 12 bytes in length, and takes the following format:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| **Int32** BlockTypeID | | | | **TimeTag** BaseTimeStamp | | | | | | | |

* BlockTypeID – The primary key of the data point description entry in a database. (revise)
* BaseTimeStamp – The reference timestamp for all data points in a particular data block.
  + TimeTag (8 bytes) – A double precision floating point number, representing the number of seconds and milliseconds elapsed since midnight on January 1, 1995.

There is one data block description for each data block in the file. The first entry in the block map refers to the first data block in the file, and so on. Recall that the footer in the file allocation table contains a field that indicates the number of data blocks, and thus the number of entries in the block map.

# Data Block

A data block holds a series of data points. Recall the following relationship:

# of data blocks = # of block map entries = file allocation table DataBlockCount field

Each data block is of a fixed length that is uniform among all data blocks in a given file. Each data block contains a series of 10 byte points.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Data Point (10 bytes) | Data Point (10 bytes) | Data Point (10 bytes) | Data Point (10 bytes) | … | Data Point (10 bytes) | Ignore (0-9 bytes) |

If the data block size is not an exact multiple of 10 bytes, the data block will have up to 9 ignored bytes at the end of the block. By convention, these bytes should be zeroes.

# Data Point

The standard data point structure is generic, and holds the value for a point referred to by the BlockTypeID field in this data block’s description entry in the block map.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Int32** OffsetTimeStamp | | | | **Short** Flags | | **Float** Value | | | |

* OffsetTimeStamp – A time offset for this data point. This offset should be added to the BaseTimeStamp for this data block to calculate the actual timestamp for this data point.
* Flags – 16 bits worth of flags
  + Undocumented
* Value – Single-precision float representation of the value for this data point.