Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
<th>Author</th>
<th>Approval</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>Initial Draft</td>
<td>Bovie</td>
<td>“Draft”</td>
<td>4/18/2014</td>
</tr>
<tr>
<td>0.2</td>
<td>Corrected inconsistent formatting</td>
<td>Bovie</td>
<td>“Draft”</td>
<td>11/18/14</td>
</tr>
</tbody>
</table>
Table of Contents
1.1 GPS Payload: ................................................................. 5
1.2 Pressure Payload: ............................................................. 5
1.3 CTD Payload: ................................................................. 5
1.4 BinAverage Payload: ....................................................... 5
1.5 BinData Payload: ............................................................ 6
1.6 Message Payload: ........................................................... 6
1.7 Vitals Payload: ............................................................... 6
1.8 Iridium CSQ Payload: ..................................................... 6
While the APEX DEEP and APF-11 store system log information in text format, they store their vital measurements and science measurements in binary formatted files named `vitals_log.bin` and `science_log.bin`, respectively. Each is composed of series of entries of different sizes per type packed end to end. The headers for each item begin with a byte-wide value indicating the total number of bytes of the entry:

```
<table>
<thead>
<tr>
<th>Size0</th>
<th>Size1</th>
<th>Size2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size0</td>
<td>Size1</td>
<td>Size3</td>
</tr>
</tbody>
</table>
```

Each item is composed of a size, data type, timestamp and payload. The size of the payload depends on the data type:

```
totalBytes: uint8_t  |  dataType: uint8_t  |  timestamp: uint32_t  |  payload: uint8_t[totalBytes-6]
```

- `totalBytes`: total number of bytes for entry, including header and payload.
- `dataType`: identifier for the payload included in the entry.
  - `LOG_VITALS_PERIODIC`: 0
  - `LOG_VITALS_IRIDIUM_CSQ`: 1
  - `LOG_SCIENCE_GPS`: 2
  - `LOG_SCIENCE_PRESSURE`: 3
  - `LOG_SCIENCE_CTD`: 5
  - `LOG_SCIENCE_BINAVERAGE_16`: 6
  - `LOG_SCIENCE_BINDATA`: 7
  - `LOG_SCIENCE_MESSAGE`: 8
  - `LOG_SCIENCE_BINAVERAGE`: 15

- `timestamp`: 32-bit Unix timestamp.
- `payload`: data type dependent value(s).

```c
#pragma pack(1)
struct log_hdr
{
    uint8_t totBytes;
    uint8_t dataType;
    uint32_t timestamp;
};
```
1.1 GPS Payload:

dataType: 2
struct gps_payload
{
    float latitude;
    float longitude;
};

1.2 Pressure Payload:

dataType: 3
struct pressure_payload
{
    float pressure;
}

1.3 CTD Payload:

dataType: 5
struct ctd_payload
{
    float pressure;
    float temperature;
    float salinity;
};

1.4 BinAverage Payload:

dataType: 6 (obsolete)
struct ctdbins16_payload
{
    uint16_t nsamples;
    uint16_t nbins;
    float max_pressure;
};

dataType: 15 (current)
struct ctdbins_payload
{
    uint32_t nsamples;
    uint32_t nbins;
    float max_pressure;
};
1.5 **BinData Payload:**
dataType: 7

```c
struct bindata_payload
{
    float pressure;
    float temperature;
    float salinity;
    uint16_t samples;
};
```

1.6 **Message Payload:**
dataType: 8

```c
struct msg_payload
{
    char msg[totBytes-6];
};
```

1.7 **Vitals Payload:**
dataType: 0

```c
struct vitals_payload
{
    float air_bladder;
    float battery;
    float humidity;
    float leak_detector;
    float vacuum;
    float coul_counter;
};
```

1.8 **Iridium CSQ Payload:**
dataType: 1

```c
struct csq_payload
{
    uint08_t csq;
};
```