IRIS-led miniSEED 3 Proposal

...

FDSN Working Group II - IASPEI 2021

Overview

- ★ Specification is complete and in FDSN web format:
  https://miniseed3.readthedocs.io/

- ★ All feedback incorporated. A direct descendent of the process, and resulting discussions, described here in 2017:
  http://www.fdsn.org/message-center/thread/514/#m-882

- ★ Specification meets WG II requirements.
  Proposal phase of the FDSN's Framework for adoption completed in 2018:
  http://www.fdsn.org/message-center/thread/527/#m-918

Simplicity of design

- Reduced complexity from miniSEED 2
- Single custom data structure
  - Except compression

Field	Description	Type	Length	Offset	Content
1	Record header indicator	CHAR	2	0	ASCII 'MS'
2	Format version	UINT8	1	2	Value of 3
3	Flags	UINT8	1	3	
	Record start time				
4a	Nanosecond (0 - 999999999)	UINT32	4	4	
4b	Year (0-65535)	UINT16	2	8	
4c	Day-of-year (1 - 366)	UINT16	2	10	
4d	Hour (0 - 23)	UINT8	1	12	
4e	Minute (0 - 59)	UINT8	1	13	
4f	Second (0 - 60)	UINT8	1	14	
5	Data payload encoding	UINT8	1	15	Data Encodings
6	Sample rate/period	FLOAT64	8	16	
7	Number of samples	UINT32	4	24	
8	CRC of the record	UINT32	4	28	
9	Data publication version	UINT8	1	32	
10	Length of identifier	UINT8	1	33	
11	Length of extra headers	UINT16	2	34	
12	Length of data payload	UINT32	4	36	
13	Source identifier	CHAR	V	40	URI identifier
14	Extra header fields	JSON	V	40 + field 10	
15	Data payload	encoded	V	40 + field 10 + field 11	

Features: requirements and more

- URI source identifier, instead of SEED codes
  - FDSN Source Identifier
- Increase sample rate to 64-bit, notation for period (to retain resolution)
- Increase start time resolution to nanoseconds
- Variable record length
- Simplifications:
  - Critical details in fixed header, fixing historical expansion
  - Fixed byte order, header is little-endian, encodings are fixed
  - Drop legacy data encodings, reserve values

Features: requirements and more

Additions:

- Format version
- Data publication version
- CRC field, allowing validation of data integrity at any point
- “Mass position off scale” flag
- “Recenter” (mass, gimbal) headers
- “ProvenanceURI” header

Optional headers in flexible, hierarchical key-value (JSON) structure:

- Specify a reserved set of extra headers defined by the FDSN, provide schema for validation
- Most previous flags and blockette contents defined in reserved extra headers
- Allow arbitrary headers to be included in a record

Forward compatibility

Near complete preservation of miniSEED 2.4 data. Information that is not retained:

- Clock model specification per timing exception
- Blockettes 400 (Beam) & 405 (Beam Delay)
- Blockette 2000 (Opaque Data)
  - *Opaque data encoding (mostly) replaces this functionality*

All other fields can be retained:

- **detailed mapping from version 2 -> 3 documented in specification**

Legacy encodings require re-encoding

Implementation Highlights

- Reference data set available
- Library to read/write format in pre-release
- Converter available
- Validator available
- IRIS fdsnws-dataselect delivering NGF

Reference data set

All time series in the reference set that contain series are the same expanding sinusoid signal.

#	Description	Download
1	Text encoded as ASCII	mseed3 JSON Text
2	Event detection headers only, no data payload	mseed3 JSON Text
3	Sinusoid series encoded as 16-bit integers	mseed3 JSON Text
4	Sinusoid series encoded as 32-bit integers	mseed3 JSON Text
5	Sinusoid series encoded as 32-bit IEEE float	mseed3 JSON Text
6	Sinusoid series encoded as 64-bit IEEE float	mseed3 JSON Text
7	Sinusoid series encoded as Steim-1 compressed integers	mseed3 JSON Text
8	Sinusoid series encoded as Steim-2 compressed integers	mseed3 JSON Text
9	Series with time quality, correction, event detections headers	mseed3 JSON Text
10	Series with some FDSN and non-FDSN extra headers	mseed3 JSON Text
11	Series with all FDSN extra headers (unrealistic)	mseed3 JSON Text

Useful for software development validation and illustrative examples

Pre-release of **libmseed** is ready

- Available in github:
  https://github.com/iris-edu/libmseed
- Full read and write support for miniSEED 2 and this specification
- Mapping from quality code to data publication version:
  - R -> 1, D -> 2, Q -> 3, M -> 4
- Not a drop-in replacement, but ease of porting was a focus

Converter available

mseedconvert:

https://github.com/iris-edu/mseedconvert

By default converts version 2 -> 3, can also convert 3 -> 2 when possible

Designed for efficient and flexible use, avoiding re-encoding when necessary

Format validator available

mseed3-validator:

https://github.com/iris-edu/mseed3-utils

Includes extra header validation using JSON Schema

Project also includes **mseed2json** and **mseed2text**, used to build reference data set.

IRIS fdsnws-dataselect delivering NGF

The DMC's fdsnws-dataselect service can deliver the new format when *format=miniseed3*

Data is converted on the fly, if needed.

Suggested next steps

- WG II conducts a Evaluation Review

- IRIS goals and intentions:
  - Prepare software ecosystem for agnostic use of either format
    - data users should not care which version they receive