Next generation miniSEED

Version 2016-3-30

Background and context

Adopted by the FDSN in 1987, the SEED format has become and still serves as the canonical format for passive source seismic (and other) data. Data exchange, especially to end users, is commonly formatted as "full" SEED, which contains both the time series and complete metadata. For continuous data collection and archiving it is common to split the time series from the metadata. Extensions to the SEED format were adopted in 1992 to define miniSEED, the time series portion of SEED, which can be decoded independently of the supporting metadata.

Many FDSN members recognize that the current two-character network code needs to expand. Such an expansion requires changes in both the metadata and time series components of the SEED format. With the adoption of StationXML by the FDSN, the metadata component is easily adjusted due to the extensibility of XML. The time series component, miniSEED, is a fixed-length field format and expanding the network code would render the format incompatible with the current release. Such a small, but disruptive change affords the opportunity to consider other changes to the format, allowing the FDSN to address historical issues and create a new foundation for current and future use.

miniSEED 3, important changes

- Expand the network code field, in coordination with equivalent StationXML changes.
  - Recommendation: 6 characters.
  - Suggested convention for temporary deployments: "xxxxYY", where 'x' are unique identifiers and YY are the last two digits of the start year of the deployment, e.g. 16 for 2016. Temporary network codes will still begin with the letters X, Y, Z, or a numeral from 0-9.
- Add a miniseed format version.
- Add a data version.
- Move most blockette 100, 1000 & 1001 field information (actual sample rate, byte order, record length, encoding, microseconds) into the fixed section of the data header.
- Simplify the record start time encoding and include microsecond resolution.
- Combine the 3 bit-flag fields in fixed section data header to a single byte, dropping rarely used flags.
- Eliminate timing correction field, timing corrections must be applied to the time stamp.
- Document forward compatibility mapping, how to convert miniSEED 2.4 to version 3.

miniSEED 3, changes for consideration

- General compression encodings for fundamental sample types and opaque data

  Encoding 50: 32-bit integers, general compressor (e.g. Brotli)
  Encoding 51: 32-bit IEEE floats, general compressor (e.g. Brotli)
  Encoding 52: 64-bit IEEE floats (doubles), general compressor (e.g. Brotli)
  Encoding 100: Opaque data
- Add CRC field for validating integrity of data payload.
- Expand channel codes, identify more instrument types and potential combination with location.
- Expand location identifier and disallow empty values (synonymous with all other series identifiers).
- Fixed-point sample encoding; would need to determine a representation due to lack of standard.
- No SEED 2.x blockettes allowed, instead allow opaque headers for arbitrary information.
- Eliminate fixed-length field for sequence numbers. Alternatives: transport protocol or opaque headers.
- Eliminate arbitrary % timing quality field, timing quality related bit flags remain. Further timing qualifiers can be in an opaque header or separate channel if needed.

Below is a straw man miniSEED Fixed Section Data Header incorporating most of the concepts above. Considerations and adjustments for byte alignment should be made after the fields have been settled.

Straw man miniSEED 3 Fixed Section Data Header

The data record starts at the first byte. The next two bytes are 'MS' to indicate the format, followed by a single binary digit indicating the format version. The fixed section of the header may be followed by optional, opaque header values. The total length of the record is the length of the fixed section, plus the length of any opaque headers, plus the length of the data payload. No padding is allowed before, after or between any of the sections.

Note	Field name	Type	Length	Offset	Mask/Flags
1	Record indicator ('MS')	A	2	-	
2	miniSEED version (3)	B	1	-	
3	Network code	A	?	-	[UN]
4	Station code	A	5	-	[UN]
5	Location identifier	A	?	-	[UN]
6	Channel codes	A	?	-	[UN]
7	Quality indicator	A	1	-	[UN]
8	Data version	B	1	-	
9	Record length	B	4	-	
10	Record start time	B	8	-	
11	Number of samples	B	4	-	
12	Sample rate	B	4	-	
13	CRC-32 of data	B	4	-	
14	Offset to data	B	2	-	
15	Flags	B	1	-	
16	Sample encoding format	B	1	-	
17	Number of opaque headers that follow	B	1	-	
18	Opaque header fields	V	V	-	

Notes for fields, all fields are mandatory:

1 Data record indicator - "MS".

2 UBYTE: miniSEED header version. Set to 3 for this version.

3 Network code. A code that uniquely identifies the network operator responsible for the data. This identifier is assigned by the FDSN. Left justify and pad with spaces (ASCII 32). Cannot be empty.

4 Station code (see Appendix G). Left justify and pad with spaces (ASCII 32). Cannot be empty.

5 Location identifier. Used to identify a grouping of channels, for example from a specific sensor. Left justify and pad with spaces (ASCII 32). Cannot be empty.

6 Channel codes (see Appendix A). Cannot be empty.

7 Quality indicator. Defined values: D (unknown), R (Raw), Q (Quality controlled), M (merged/modified).

8 UBYTE: Data version. Start with version 1 and increase for later versions.

9 ULONG: The record length in bytes.

10 LONGLONG (64-bit signed integer): Start time of record, time of the first data sample. As a representation of UTC, this value is encoded as the number of microseconds since midnight 1 January 1970 UTC not including leap seconds. This is a microsecond version of Unix/POSIX time as defined by IEEE Std 1003.1, 2013 Edition (POSIX.1-2008). The mapping between separate components of a UTC time (seconds, minutes, hours, etc.) and this representation is documented in Section 4.15 of IEEE Std 1003.1, 2013 Edition, which is then scaled by 1E6 and microseconds are added to result in this representation. This time scale is continuous except for the occurrence of leap seconds, whether this value is a leap second or not is defined by bit 2 of the Flags field. When calculating time within a record, bits 2 and 3 of the Flags field should also be consulted to determine if leap seconds occurred during the record.

11 ULONG: Number of data samples in record.

12 FLOAT: Sample rate encoded in IEEE-754 floating point format. When the value is positive it represents the rate in samples per second, when it is negative it represents the sample period in seconds. Writers should use the negative value sample period notation for rates less than 1 samples per second to retain resolution. Set to 0.0 if no time series data is included or data is opaque.

13 ULONG: CRC-32 value of data as defined and used in RFC 1952 (GZIP format). For non-opaque data this is the CRC value of the decoded data payload. For opaque data it is the CRC of the raw payload. If no data payload or a CRC is not possible, set this value to 0.

14 UWORD: Offset in bytes, relative to the beginning of the record, to the beginning of encoded data. If no data payload, set this value to 0.

15 UBYTE: Flags:
[Bit 0] - Byte order. Set this bit to 0 to indicate least significant byte first (little endian) order and 1 to indicate most significant byte first (big endian) order. This indicates the byte order of binary header and data samples values.
[Bit 1] - The start time occurred during a leap second.
[Bit 2] - A positive leap second occurred during this record.
(same as SEED 2.4 FDSN, field 12, bit 4)
[Bit 3] - A negative leap second occurred during this record.
(same as SEED 2.4 FDSN, field 12, bit 5)
[Bit 4] - Time tag is questionable. (same as SEED 2.4 FSDH, field 14, bit 7)
[Bit 5] - Clock locked. (same as SEED 2.4 FSDH, field 13, bit 5)

16 UBYTE: A code indicating the encoding format. (same as SEED 2.4 Blockette 1000 field 3, with addition of encodings 50, 51, 52 and 100 described above)

17 UBYTE: Total number of opaque header fields that follow the fixed section.

18 VAR: Opaque data header fields. Each opaque header field is a variable length string, terminated by the character '~' (ASCII 126). Each header may contain any data except for the terminating character. It is strongly recommended that opaque headers contain printable text. Example header values (with terminators), for illustration only, no implied usage pattern:
"GPS~", "TYPE=GPS~", "FORMAT=BINEX~", "SEQUENCE=12345~", "FILENAME=data.bin~", "FRAGMENT=15/238~", "TIMEQUALITY=98%~"