Marine seismology data/metadata standards

1 Introduction

1.1 Nomenclature

This document contains FDSN standards, proposed standards, and recommendations.

- Proposed standards are preceded by "**We propose**".
- Recommendations are preceded by "**We recommend**".
- FDSN standards are stated.

If an element name has a defined unit, the unit is specified as <name>.<unit>.

1.2 A general rule for adding marine-specific information to StationXML

We propose the following rule:

- Define the sub-elements that need to be specified
- Insert into StationXML as a StationXML-standardized element

- Request the addition of the information into StationXML, through FDSN WGII

1.3 StationXML-standardized element

Is an element that obeys the StationXML schema but contains information that is not specified in the schema.

Currently, StationXML-standardized elements are encoded as StationXML `<Comment>`s, with the elements written as JSON-formatted strings, preferably with a marine-specific `subject`. For example, if the information concerns "peanuts" and can be expressed as:

```
can_size.ml: 150
num_nuts: 10
brand_name: "Mr Peanut"
nut_weights.g: [1.1, 1.1, 2.2, 2.4, 1.8, 1.4, 1.3, 1.2, 3.0, 2.1]
```

then this would be entered into the StationXML file (at the station or channel level, as appropriate) as:

```
<Comment subject=”peanuts”><Value>“{can_size.ml: 200. num_nuts: 10, brand_name: "Mr
Peanut", nut_weights.g: [1.1, 1.1, 2.2, 2.4, 1.8, 1.4, 1.3, 1.2, 3.0,
2.1]}”</Value></Comment>
```

1.4 Geophone wiring and polarity

JCollins:

1.5 Specifying and correcting drift in data that contains a leap second

- **We propose** that the instrument clock drift times provided in the stationXML file be leap-second corrected
- **We propose** that NOT CLOCK CORRECTED data not be leap-second corrected, and that CLOCK CORRECTED data be leap-second corrected
- ***We propose*** *that the drift correction be performed in two steps:*
    - *OPTION 1 (*
        - *If you use the data to calculate the drift rate, first generate LEAP-CORRECTED data from the NOT CLOCK CORRECTED DATA. Use this data to calculate the clock drift. Throw the LEAP-CORRECTED data away.*
        - *Next, generate CLOCK CORRECTED data from the NOT CLOCK CORRECTED data by first applying the drift correction, then the leap-second correction. This avoids the possibility that the record containing the leap-second will change, which would generate an extra gap and overlap in the data. The offset it will create in the time correction is of the order of the drift rate * 1 second, which will be below 1 microsecond for a drift rate below 1e-6 (< 31 seconds/year)*
    - *OPTION 2:*
        - *First, integrate the leap-seconds. Verify that this adds no gaps or overlaps*
        - *If you use the data to calculate the drift rate, do it here.*
        - *Next, apply the drift correction*
        - *Finally, validate that there are still no added gaps/overlaps. If there are, remove all leap-second flags and re-run leap-second insertion*

2 Proposed standards and recommendations

2.1 Marine-specific

2.1.1 StationXML

2.1.1.1 Clock drift and leapseconds

There is no dedicated field in StationXML. Embed this information as JSON-coded strings JSON-coded in Station-level `<Comment>` fields with `subjet="Clock Correction"`. In the future, a seperate namespace may be created to allow a more specific and structured representation

The structures are shown below in YAML format first, for clarity, then as a StationXML-standardized Element.

2.1.1.2 Clock drift

We recommend specifying clock drifts using UTC datetimes, to avoid ambiguity. The datetimes should be in ISO8601 format, followed by a "Z" to unambiguously specify UTC.

2.1.1.2.1 YAML format:

```
drift:
    type: 'piecewise_linear'  # or 'cubic_spline' or 'polynomial {a0 a1 a2 ...}'
    instrument: 'Seascan MCXO'
    instrument_nomimal_drift_rate: 1e-8
    reference: 'GPS'
    syncs_instrument_reference:
        - ["2016-09-10T00:00:00Z", "2016-09-10T00:00:00Z"]
        - ["2017-01-12T00:00:01Z", "2017-01-12T00:00:00.415Z"]
        - ["2017-07-13T11:25:01Z", "2017-07-13T11:25:00.6189Z"]
```

2.1.1.2.2 StationXML format:

```
<Comment subject=”Clock Correction”><Value>“{drift: {instrument: Seascan MCXO,
 instrument_nominal_drift_rate: 1e-8, reference: GPS, type: piecewise_linear,
 syncs_instrument_reference: [['2016-09-10T00:00:00Z', '2016-09-10T00:00:00Z'],
 ['2017-01-12T00:00:01Z', '2017-01-12T00:00:00.415Z'], ['2017-07-13T11:25:01Z',
 '2017-07-13T11:25:00.6189Z',]]}}”</Value></Comment>
```

2.1.1.2.3 Explanation of fields

The `time_base`, `nominal_drift_rate` and `reference` fields are optional. In the simplest case of a synchronization at the start and end of an experiment, and assuming purely linear drift, there would only be two items in syncs_instrument_reference.

2.1.1.2.4 If no drift is measured

Specify:

```
<Comment subject=”Clock Correction”><Value></Value></Comment>
```

This means that there is a clock drift, but we don’t know what it is

2.1.1.3 Leap seconds

Specified using information from the `leap-seconds.list` file, available online at several sites, including https://data.iana.org/time-zones/tzdb/leap-seconds.list. The user should verify that the "File expires on" date is later than the last instrument channel's end-date.

2.1.1.3.1 YAML format:

```
leapseconds:
    list_file_entries:
        - line_text: "3692217600      37      # 1 Jan 2017"
          leap_type: '+'
```

2.1.1.3.2 StationXML format

```
<Comment subject=”Clock Correction”><Value>{list_file_entries: [{
 line_text: '3692217600      37      # 1 Jan 2017', leap_type: '+'}]}</Value></Comment>
```

2.1.1.3.3 Explanation of fields

- `list_file_entries` is an array/list, to allow for more than one leap-second during a deployment.
    - `line_text` should be directly copied from the `leap-seconds.list` file
    - `leap_type` indicates whether the 2nd number in the list_file line_text ("37" in the above example) is greater than the previous line's value ("+") or less than the previous line's value ("-"). As of June 2024, all leap seconds have been type "+"

2.1.1.3.4 If the not clock-corrected data integrates leapsecond information

Add `leapsecond_is_integrated_into_not_clock_corrected_miniseed: True` to the StationXML Comment:

```
<Comment subject=”Clock Correction”><Value>{list_file_entries: [{
 line_text: '3692217600      37      # 1 Jan 2017', leap_type: '+'}],
leapsecond_is_integrated_into_not_clock_corrected_miniseed: True}</Value></Comment>
```

2.1.1.4 Orientation information

Set the following `<Azimuth>` and `<Dip>` values for the following source/subsource codes:

code	<Dip unit="DEGREES>xxx</Dip>[1]	<Azimuth unit="DEGREES" xxx>	yyy</Azimuth>	Comment
1	0.0	minusError="180.0" plusError="180.0"	0.0	Equivalent “N” for non-geographically oriented
2	0.0	minusError="180.0" plusError="180.0"	90.0	Equivalent “E” for non-geographically oriented (90° clockwise of “1” when viewed from above)
3	90.0		0.0	Positive voltage for DOWNWARD motion
N	0.0		0.0	Azimuth must be within 5° OF 0°
E	0.0		90.0	Aziumth must be within 5° OF 90°
Z	-90.0		0.0	Positive voltage for UPWARD motion. Dip must be within 5° of -90
DH, DG, DO	90.0		0.0	if value *DECREASES* for a pressure increase[2]

[1] StationXML/Seed Dip is degrees down from horizontal. SAC "CMPINC" is StationXML Dip+90 degrees (Component incident angle (degrees from upward vertical))

[2] The pressure sensor dip gives the same polarity as the seismometer/geophone for UPGOING waves. Dip = -90 means that the first break will have the same polarity as a "Z" channel

code	<Dip unit="DEGREES>xxx</Dip>[1]	<Azimuth unit="DEGREES" xxx>	yyy</Azimuth>	Comment
DH, DG, DO	-90.0		0.0	if value *INCREASES* for a pressure increase

2.1.1.5 Data completeness

We recommend using Station <StartDate> and <EndDate> to specify when the data was supposed to start and end, and Channel <StartDate> and <EndDate> to specify when it actually starts and ends for each channel.

We recommend keeping all of the recorded data, including "noisy" or "bad".

2.1.1.6 Leveling system

The instrument's leveling system can affect the quality and absolute values of measurables. The following elements should be specified:

```
threshold.deg: (float) # deviation from vertical (degrees) which will trigger releveling

accuracy.deg:  (float) # maximum deviation (degrees) from vertical accepted after relevel

max_relevel_interval.h (float) # longest interval between level checks during the deployment (hours)

n_relevels: (int) # number of relevels performed during the deployment

relevels: (list) # list of [date, level_before.deg, level_after.deg] for each relevel

description: (str) # description of the leveling system (ex: “gravity-based gimbal system in oil”, or “Guralp Aquarius automatic compensation system”). Manufacturer-defined description should be used if available.
```

We propose implementing this as an "Equipment" element at the appropriate (Station or Channel) level, with <Type>Leveler</Type> and <Description> a JSON-encoded string of the above elements.

We will request a specific ``Leveler`` implementation of the ``Equipment`` element, with the above elements added to it.

2.1.2 subnetwork files

We recommend using obsinfo subnetwork files to store essential information about OBS deployments.

obsinfo subnetwork files can be used with the obsinfo software to generate FDSN StationXML files with embedded OBS-specific information, or with other tools to modify existing StationXML files.

2.1.3 miniSEED data

2.1.3.1 Clock drift correction

2.1.3.1.1 Overview

There are three main possibilities for distributing data:

1. **"NOT CLOCK CORRECTED"**: No time correction applied. May be preferred by users of long-period data (>10s) because it can be easier to concatenate daily files.
    a. **We propose** for Indicating:
        i. miniSEED2: Set data quality flag "D"

ii. miniSEED3: Set root -> FDSN -> DataQuality to “D”

b. **We recommend** for Creating:

i. miniSEED2: Put time correction in record header field 16 and set field 12 bit 1 to 0.

ii. miniSEED3: No specific header information

2. **"CLOCK CORRECTED"**: Indicate the time correction in each record header and apply it. Allows the user to work with time-corrected but otherwise unmodified data.

a. **We propose** for Indicating:

i. miniSEED2: Set data quality flag to "Q"

ii. miniSEED3: Set root -> FDSN -> DataQuality to “Q”

b. **We recommend** for Creating:

i. Calculate a new time drift for each record

ii. miniSEED2:

1. In record header field 16 (“Time Correction”), specify correction added to original time (in units of 0.0001 seconds).

2. Set record header field 12, bit 1 (“Activity Flag, time correction applied”) to 1.

iii. miniSEED3:

1. In root -> FDSN -> Time -> Correction, specify correction added to original time (text, in units of float seconds)

3. **"RESAMPLED"**: Resample the data at the originally intended rate. Date are time-corrected and easy to concatenate/combine with other data, but could distort waveforms/spectra (needs study).

a. **We recommend** for Indicating:

i. miniSEED 2/3: Define another channel code (modified data)

b. **We recommend** for Creating

i. No recommendation

2.1.3.1.2 Recommendations

We recommend always providing at least "**CLOCK CORRECTED**” data, if possible.

We recommend, if there is no measure of clock drift:

- miniSEED2:
  - Provide data as "D". set bit 7 of data quality flag ("time tag is questionable") to 1.
  - Add blockette 500, field 10 ("Clock status") indicating that there is an unmeasured drift (for example: "Unmeasured clock drift on Seascan MCXO, expected order = 1e-8")
- miniSEED3

- No recommendation yet.

A new ``msmod``, in development, should be able to create **CLOCK CORRECTED** data from **NOT CLOCK CORRECTED** data, using piecewise linear, cubic spline, or polynomial interpolation.

2.1.3.1.3 Providing multiple data types

Most FDSN-compatible data centers store data in SeisComP Data Structure (SDS), which does not distinguish between data with different data qualities. If you want to provide **CLOCK CORRECTED** and **NOT CLOCK CORRECTED** data, they must share the same files.

2.1.3.2 Leap seconds

Leap seconds should be corrected in **CLOCK CORRECTED** data and the record containing the leap second should be flagged.

2.1.3.2.1 Positive leap second

61 seconds in the minute .

This is the only case as of year 2025:

- Shift all record times AFTER the leap second back one second.
- Set activity flag bit 4 to 1 in the header of the record containing the leap second.
- Change `end_sync_instrument` to be one second earlier than what the instrument indicated

2.1.3.2.1.1 Example

Changing the data using msmod, for a positive leap-second at 23:59:60 on day 182, 2016:

```
msmod --timeshift -1 -ts 2016,182,23:59:59.999999'
msmod –-actflags ‘4,1’ –tsc 2016,182,23:59:59.999999 –tec 2016,182,23:59:59.999999
```

A new ``msmod`` option has been proposed to simplify this to one line and one time specification.

2.1.3.2.2 Negative leap second

59 seconds in the minute

There have been no negative leap seconds as of 2025.

- Shift all record times AFTER the leap second forward one second.
- Set activity flag bit 5 to 1 in the header of the record containing the leap second.
- Change `end_sync_instrument` to be one second later than what the instrument indicated

2.2 Marine (and maybe general, too)

2.2.1 processing steps

We recommend that processing done on data files (from data download to delivery to the data center) should be recorded in text-based, structured files. The JSON process-steps format is an example.

2.2.2 Proposed modifications to existing standards

2.2.2.1 Allow sampling rate to be specified as double precision[3]

This is the only way to accurately represent OBS clock rates, which are regular but off of the specified sampling rate by a factor of approximately 1e-8 (MCXOs) or 1e-9.5 (CSACs), requiring 27- or 32-bit

[3] Double precision sampling rates are included in the miniSEED3 specification

floating-point mantissas, respectively, to be correctly specified. Single precision floats only have 23-bit mantissas, double precision floats have 52-bit mantissas.

2.2.2.1.1 In miniSEED3

In the miniSEED3 header, the Data publication version replaces the data quality flags (can still have flags in “Extra Header Fields” (field 14). This offers a clear hierarchy, but not a way to specify that one wants uncorrected or corrected data (recommended RAW=1 could be used for uncorrected data). Could this be put in field 14: extra header fields? In any case, would have to be searchable using web tools

3 Reminder/clarification of existing standards

3.1 Source Identifiers

The following source-subsource codes (see FDSN Source Identifiers documentation) should be used for the following types of sensor/data:

Code	Description
1	Unoriented seismometer, “N” channel equivalent
2	Unoriented seismometer “E” channel (+90 degrees from "1")
3	Seismometer/geophone with inverted vertical channel (positive voltage is down)
DH	Hydrophone
DG	Differential pressure gauge
DO	“Absolute” bottom pressure recorder

3.2 Station names for repeated deployments

The IASPEI Station Coding Standard recommends that station and/or location codes be changed if the associated sensors are moved far enough to result in a significant *teleseismic* travel-time residual discrepancy. We recommend the same, except that the basis for what is a significant travel-time residual discrepancy should depend on your study.

If you change the station name between deployments, we recommend incrementing the last *N* characters of the station name, e.g. ``STAA``, ``STAB``, ``STAC``, or ``STA01``, ``STA02``, ``STA03``, etc. The value of *N* depends on the maximum number of deployments you will possibly make and the characters you use in the incrementor (numeric, alphabetic, or alphanumeric).

3.3 Deployments in lakes

Set `<WaterLevel>` to the elevation of the lake surface

3.4 Positions

We recommend using the `plusError`, `minusError` and `measurementMethod` attributes to specify uncertainties in Latitude, Longitude and Elevation and how you measured them.

3.5 Standard values that marine seismologists may not know:

Within each `<Channel>`, set `<Type>CONTINUOUS</Type>` and `<Type>GEOPHYSICAL</Type>`

For pressure sensor channel responses, `InputUnits` should be `Pa`, without a `Description`, i.e.:

```
<InputUnits><Name>Pa</Name></InputUnits>
```