The FDSN WG5 Action Group on Marine Seismology Standards

Wayne C. Crawford[1], Kasey Aderhold[2], Yinshuang Ai[3], Jerry Carter[2], John A Collins[4], Carlos Jorge Corela[5], Susanne Hemmleb[6], Takehi Isse[7], Joel D. Simon[8], Maria Tsekhmistrenko[9]

[1]IPGP, CNRS & Univ. Paris Cité; [2]Earthscope Consortium; [3]Inst Geol. & Geophys., Chinese Acad. Of Sci.; [4]Woods Hole Oceanographic Inst.; [5]Inst. Dom Luiz, Fac. Ciencias da Univ. Lisboa; [6]Helmholtz Centre Postdam GFZ German Research Center for Geosci.; [7]Univ. Tokyo; [8]Princeton Univ.; [9]Univ. College London

Overview

- Goal: Definition of standards for marine seismology data and metadata
- Time Period: Sept 2023-present (2 years, bimonthly meetings)
- Members: 4 North America, 4 Europe, 2 Asia
- Subjects:
  - StationXML standards
  - miniSEED standards
  - Source identifier standards
  - Software

Major Issues

- Clock drift
- Leapseconds
- Data completeness
- Leveling systems
- Positions and Orientations

Outputs

- Recommendations, propositions and reminders
  - StationXML
  - miniSEED
  - Source identifiers
  - other
- Unresolved issues
- Proposed project for software integration

Source Identifiers

- **Recommendations**
  - if there are two orthogonal horizontal channels that are not known to be aligned to N and E, they be named 1 and 2, with 2 oriented 90° clockwise of 1 when viewed from above.
  - If the vertical channel is inverted, name it “3”
- **Proposals**
  - If a component’s angle relative to another component is known, but it’s absolute angle is not, the given angle should be the relative angle, and “plusError” and ”minusError” should be set to 180.

StationXML metadata

- StationXML-standardized element:
  - A marine-specific element that has no StationXML equivalent. These are currently:
    - Clock drift specification
    - Leapsecond information
    - Leveler specification
  - Expressed as a Comment or an Equipment with the marine-specific subelements expressed as a JSON string in the Value or Description, respectively
  - If a subelement has a defined unit, it is specified as <name>.<unit>, with the unit in UCUM nomenclature.

StationXML metadata

- **Clock drift** StationXML-standardized element
  - Station-level Comment with:
    - Subject="Clock Correction"
    - Value=JSON-string of

```
{drift: {
    type: {"piecewise_linear", "cubic_spline" or "polynomial {a0} {a1} {a2}…"},
    instrument: <str>,
    instrument_nominal_drift_rate: <float>,
    reference: <str>
    syncs_instrument_reference: [["<datestr>", "<datestr>"], ["<datestr>",
"<datestr>"], …]
}}
```

  - Example:

```
<Comment subject="Clock Correction"><Value>{drift: {type: {"piecewise_linear"},
instrument: 'Seascan MCXO',instrument_nomimal_drift_rate: 1e-8,reference: 'GNSS',
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>
```

  - The instrument sync times should be leapsecond-corrected, if applicable

StationXML metadata

- **Clock drift** StationXML-standardized element
  - If no drift measured but we expect drift:

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

StationXML metadata

- **Leapsecond** StationXML-standardized element
  - Network-level Comment :
    - Subject=“Clock Correction”
    - Value=JSON-string of

```
{leapseconds: {
    list_file_entries: [
        [{line_text: <str>, leap_type: ”+” | “-”}], …]
}}
```

    - Example:

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

  - Station-level Comment if NOT_CLOCK_CORRECTED data is leapsecond corrected:

```
<Comment subject=“Clock Correction”><Value>{leapseconds:
{leapsecond_is_integrated_into_not_clock_corrected_miniseed: True}}</Value></Comment>
```

StationXML metadata

- **Leveling system** StationXML-standardized element
  - Channel-level Equipment element with :
    - Type="Leveler"
    - Value=JSON-string of

```
{threshold.deg: <float>,
 accuracy.deg: <float>,
 max_relevel_interval.h: <float>,
 n_relevels: <int>,
 relevels: [{date: "<datestr>", level_before.deg: <float>, level_after.deg: <float>},
            ...],
 description: <str>
}
```

    - Example:

```
<Equipment><Type>Leveler</Type><Description>{threshold.deg: 2.5, accuracy.deg: 0.25,
max_relevel_interval.h: 168, n_relevels: 12, description: 'SIO-Nanometrics BBOBS
leveling system'}</Description><Manufacturer>Scripps Inst.
Oceanography</Manufacturer><SerialNumber>FR02</SerialNumber></Equipment>
```

StationXML metadata

- **Leveling system** proposal
  - Channel-level EquipmentType element named Leveler with added Leveler-specific subelements

StationXML metadata

- Positions and Orientation
  - **We recommend** setting the `plusError`, `minusError` and `measurementMethod` sub-elements to specify uncertainties in `Latitude`, `Longitude, Elevation, Dip and Azimuth`, and how these values were estimated.
- Deployments in lakes
  - **We remind** that the `<WaterLevel>` element should be set to the elevation of the lake surface
- 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".
- **Reminders** of standard values:
  - ??Within each `<Channel>`, set `<Type>CONTINUOUS</Type>` and `<Type>GEOPHSYSICAL</Type>`??
  - For pressure sensors, set `<InputUnits><Name>Pa</Name></InputUnits>` (no Description)

miniSEED data

- Leap second correction
  - Using msmod

miniSEED DATA

- The correction is the seconds to add to the original time

	miniSEED2	miniSEED3
NOT CLOCK CORRECTED	Quality flag = 'D' Field 12 bit 1 = 0	Root/FDSN/DataQuality = 'D'
CLOCK CORRECTED	Quality flag = 'Q' Field 16 = correction in 0.00001s Field 12, bit 1 = 1	Root/FDSN/DataQuality = 'Q' Root/Time/Correction – correction in s
Unmeasured drift	Quality flag = 'D' Field 14, bit 7 = 1 Blockette 500, field 10 = "unmeasured clock drift, expected order = 10^-XX"	Root/FDSN/DataQuality = 'D' ?????

- **RESAMPLED data** would needs another channel code to specify modified data (none currently in Source Identifiers)

Other information

- Remind that if CLOCK CORRECTED and NOT CLOCK CORRECTED are provided in SDS format, they must be in the same file
- Recommend providing CLOCK CORRECTED
- Order in which to provide leap second and clock drift corrections
- How to request marine-specific additions to StationXML
- New msmod version
- Saving processing steps information
- Station names for repeated deployments
- Geophone wiring and polarity

Open issues

- Dataloggers that correct leap seconds
- Dataloggers that output StationXML metadata
  - List and conformity
- StationXML standardized element using namespace
- Optional methods for applying clock drift + leapsecond corrections
- Should the miniSEED3 sampling rate include the clock drift?
- Msmod updates
  - Clock correction
  - Simplified leapsecond correction

Simplified leapsecond correction

- Current (positive leap second)

```
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
```

- Proposed

```
msmod --lsp 2016,183,00:00:00
```

- Add ' --lsp ' and ' --lsn ' options, might be able to remove ' --tsc ' and ' --tec'

To Do

- Tie up loose ends, based on feedback from WP2
- Decide how/where the above should be documented
- Document on official FDSN sites
  - Source identifiers (?)
  - StationXML
  - miniSEED3
  - Marine-specific document?

Marine-specific software

- Currently have a document with a list of known software
- Propose to
  - Create an FDSN github site with wiki to allow users to add other software

2 Available software

Subject	Software	Comments
Modifying miniseed files	msmod	Adding leapseconds into data, modifying header information (network, station, location and channel codes, dataquality code, etc)
	GIPP tools	Toolbox: modify header, cut in pieces, export header to ASCII, etc.
Creating metadata	obsinfo	Create stationXML files from instrumentation files and a "subnetwork" file describing deployments with marine-specific information (Crawford et al., 2025). Can also create yasmine-stationxml-cli commands to inject marine-specific information into stationXML files.
	yasmine-stationxml-editor	Python web application to create and edit stationXML files. Does not include marine-specific information.
	yasmine-stationxml-cli	Command-line script for merging/editing stationXML files from online instrumentation databases.
Calculating clock drift	Integrated Seismic Program	Graphical user interface within Ambient Noise toolbox (Cabieces et al., 2022). Also includes toolboxes for receiver functions, time-frequency analysis, earthquake analysis and moment tensor inversion
	OCloC	Python package associated with Naranjo et al. (2024).
Correcting clock drift	qedit	Linear and piecewise linear clock drift correction
Calculating sensor orientation	ppol	P-wave polarization, based on Scholz et al. (2017), plus event location uncertainty
	OBSIP orientation	Uses Rayleigh wave polarization (Stachnik et al., 2012)
	OrientPy	Audet (2020). Includes Rayleigh-wave polarization (Doran and Laske, 2017) and P-wave polarization (Braunmiller et al., 2020).
Removing noise	ATACR	Automated Tilt and Compliance Removal (Janiszewski et al., 2019). Matlab version
	ATACR	Python version
	tiskitpy	Separate transfer-function based and rotation-based classes.
	BRUIT-FM Toolbox	Improved windowing selection and data stacking for more accurate transfer function calculation (Rebeyrol et al., 2024)
	Compy	Based on tiskitpy, adds periodic glitch removal and data processing automation

Marine-specific software

- Propose to
  - Create an FDSN github site with the above list and wiki to allow users to add other software
  - Create an Action Group to create
    - Validated software
    - Rules for validate software
      - Documentation
      - Installation
      - Verified on Linux + Windows + macOS + ???
    - Integrated software? (need to hire software engineer)