OBS data/metadata proposal

Wayne Crawford

Version: 201910

Version	Modifications
201805	Went back to recommendation of data quality flag for specifying whether time-corrected or not (allows semi-automatic selection using FDSN webservices, does it work for ArcLink?)
201809	Added request for versioning Requested more quality control flags. Changed recommended azimuths for horizontal channels 1 and 2 from 0 and 0 to 0 and 90 degrees (with 180 degree uncertainties). Changed absolute pressure gauge channel naming to match OO
201906	Clarified channel <Azimuth> and <Dip> values, adding hydrophone channel
201910	Fixed an error in the 'H' channel dip value

Preamble

Ocean bottom seismometers (OBS) provide seismological access to the 70% of the earth's surface that is covered by water. There are at least OBS experiments/year around the world, of which less than 50% are saved in FDSN-compatible seismological databases. The unsaved data often has incomplete metadata and is therefore hard to recover later on. The distribution of data by OBS facilities is hampered by a lack of clear standards and procedures for archiving data and metadata.

We propose here 1) modifications to the stationXML and miniSEED formats that will allow OBS (and other) data to be better informed; 2) OBS-specific standards and "best practices" for using the stationXML and miniSEED formats to make OBS data the most clear and easy to use; 3) post-processing tools for OBS data that should be made available to seismologists in order to reduce OBS-specific problems and take advantage of OBS-specific data possibilities.

The document is divided into three sections: 1) Proposed Modifications to StationXML and miniSEED formats; 2) OBS data/metadata standards and best practices; 3) Recommended OBS-specific Software

Proposed modifications to the StationXML and miniSEED formats

StationXML

Add a "Water level" field

This is useful for removing/exploiting water surface reflections. In general, this would be set to 0 (sea level), but would be different if deployments are made in lakes or water-filled boreholes

We chose "water level" rather than "water depth" because the default value would be "0" rather than "-elevation".

Add a "Localization method" field

This text field would indicate how the sensor position was determined. Probably often "GPS", but could also be something like "Laser-based distance and compass-based angle from location 00" or "Acoustic survey", etc.

Option: Add a "Measurement_method" attribute to uncertaintyDouble. That would allow one to also specify, for example, how Azimuth and Dip were determined.

Add a "CommentList" type

Would allow several related comments to be grouped together. Similar to the <Comment> type except that a <Subject> field would be added and <Value> would be changed to <Values> or <List> with multiple strings allowed

Allow versioning

Some mechanism for specifying the version (perhaps with a means of specifying changes between versions)

miniSEED

Allow sampling rate to be specified as double precision.

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

Allow versioning

As with metadata.

More data quality flags, with clear hierarchy

Data quality flags are the only clear way to distinguish between levels of data processing, but the choices are too limited. Additional data qualities that cannot currently be specified are: Data directly translated from another format , or data for which the header values have been changed, but not the data itself. A possible hierarchy would be (new in italics):

- "D" : The state of quality control of the data is Indeterminate
- *"T": Translated Raw Waveform Data from another initial format*
- "R": Raw Waveform Data with no Quality Control (reserved for SEEDlink)
- *"H": Quality controlled Data, processes have been applied only to the headers*

- "Q": Quality controlled Data, some processes have been applied to the data *(does this mean time-series values)?*
- *"C": Quality controlled Data, No processes applied to time-series or header*
- "M": Data center modified, time-series values have not been changed

OBS data/metadata standards and best practices

Timing corrections

OBS clocks generally have a non-negligible drift because of the lack of GPS signal at the seafloor. The resulting time offsets must be corrected or at least indicated in any data archived at data centers. OBS time bases are generally chosen to have small and first-degree linear drift. Their drift is calculated by synchronizing the instrument clock to GPS before the deployment and then comparing the instrument clock to GPS after the deployment. If the instrument clock cannot be compared to GPS at the end of the experiment, the drift can be calculated a posteriori by calculating the noise correlation between this instrument and another synchronized instrument over the length of the experiment.

Information about the existence of linear clock drift, its value if measured and its probable range if not measured, should be provided in the data and metadata.
We recommend the following practices:

StationXML

Indicate the timing correction in <Comment> or <CommentList> fields, as follows:

```
<CommentList>
    <Subject>Linear Clock Correction</Subject>
    <List>
        <Value>“time_base: Seascan MCXO, ~1e-8 nominal drift"</Value>
        <Value>“reference: GPS"</Value>
        <Value>“start_sync_reference: 2015-04-22T09:21:00Z"</Value>
        <Value>“start_sync_instrument: 0"</Value>
        <Value>“end_sync_reference: 2016-05-28T22:59:00.1843Z"</Value>
        <Value>“end_sync_instrument: 2016-05-28T22:59:02Z"</Value>
    </List>
</CommentList>
```

If the <CommentList> modification is not accepted, bundle the same in a <Comment>, using JSON syntax:

```
<Comment>
    <Value>“{Linear Clock Correction: {time_base: Seascan MCXO, ~1e-8 nominal
drift, reference: GPS, start_sync_reference: 2015-04-
22T09:21:00Z,start_sync_instrument: 0, end_sync_reference: 2016-05-
28T22:59:00.1843Z,end_sync_instrument: 2016-05-28T22:59:02Z}}”</Value>
</Comment>
```

Absolute dates are used because they are unambiguous. “drift” or “slew” values are derived values and there is no standard for whether a positive value means the instrument is faster than GPS or vice versa

miniSEED

There is no consensus yet on whether/how to apply calculated time corrections. Three main possibilities have been proposed:

1. Indicate the time correction in each record header but do not apply it (RAW).
2. Indicate the time correction in each record header and apply it (SHIFTED).
3. Resample the data at the originally intended rate (RESAMPLED)

The author prefers the SHIFTED method as it allows the user to work with time-corrected data which has not been modified but for which the time is as close as possible to GPS time. Until consensus is reached, we propose below how to distinguish between these methods.

If the time correction has been calculated:

- RESAMPLED data: Use a non-standard Instrument Code, as the data themselves have been modified.
- SHIFTED data:
    - Indicate time correction applied in record header field 16 ("Time Correction" and set field 12, bit 1 ("Activity flag, time correction applied") to 1. The 'qedit' software does all of these at once to each header ("add_trend corr" and "apply_corr keep" commands).
    - Indicate that the time correction code has been applied by:
        - Setting the data quality flag to "Q"
        - Alternatively, specify a location code between 00 and 49
- RAW data.
    - Indicate time correction applied in record header field 16, without applying it. The 'qedit' software can do this using its "add_trend corr" command.
    - Indicate that there is no time correction by:
        - Setting the data quality flag to "D
        - Alternatively, specify a location code between 50 and 99

If the time correction has not been calculated

Set bit 7 of the data quality flag ("time tag is questionable") to 1.
?If possible, add blockette 500, field 10 ("Clock status") indicating the linear drift (i.e. "Unmeasured linear drift on Seascan MCXO, expected order(1e-8)")?

Leap seconds

Leap seconds should be corrected in the data and the record containing the leap second flagged. It may also be useful to state in the StationXML how the correction was made?

miniSEED

If the leap second is positive (the most common case: 61 seconds in the minute):

- 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

If the leap second is negative (59 seconds in the minute):

- 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

Here is how to do this using msmod, assuming 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' –ts 2016,182,23:59:36 –te 2016,183,00:00:36
```

The times in the second command are hand-chosen to bracket the record containing the leap second.

StationXML

?Add a comment (or CommentList) with the leap second information?, for example:

```
<CommentList>
    <Subject>Leap Second Correction</Subject>
    <List>
        <Value>“time: 2016-082T23:59:60Z"</Value>
        <Value>“description: Positive leap-second (a 61-second minute)"</Value>
        <Value>“correction_data:
               msmod --timeshift -1 -ts 2016,182,23:59:59.999999 –s”</Value>
        <Value>“correction_end_sync_instrument:
               subtracted one second from displayed instrument time"</Value>
</List>
</CommentList>
```

Orientation information

Set the following <Azimuth> and <Dip> values for orientation codes 1, 2 and 3:

1	<Dip unit="DEGREES">0.0 </Azimuth><Azimuth minusError="180.0" plusError="180.0" unit="DEGREES">0.0</Azimuth>
2	<Dip unit="DEGREES">0.0 </Azimuth><Azimuth minusError="180.0" plusError="180.0" unit="DEGREES">90.0</Azimuth>
3	<Dip unit="DEGREES">-90.0 <Azimuth unit="DEGREES">90.0</Azimuth>
H[1]	<Azimuth unit="DEGREES">0.0</Azimuth> <Dip unit="DEGREES">-90.0</Dip>

StationXML

Use Station <CreationDate> and <TerminationDate> fields to specify when the data was supposed to start and end, and <StartDate> and <EndDate> to specify when it actually starts and ends.

Standard values you may not know:

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

miniSEED

Use Steim1 compression if possible (almost as compact and more reliable than Steim2)

Channel Polarity

All data should have GSN-standard polarity (including positive Z = vertical motion “up”). Geophones generally give positive voltages DOWN, so their signal should be INVERTED within the OBS or their channel named “3”[2]

[1] Assumes the hydrophone has positive voltage for a positive (compressional) pressure: if it has negative voltage for a positive pressure, then Dip should = 90.0

[2] J Clinton prefers “1”, with horizontals = “2” and “3”, as is done for boreholes, but this goes against IRIS OBS channel nomenclature

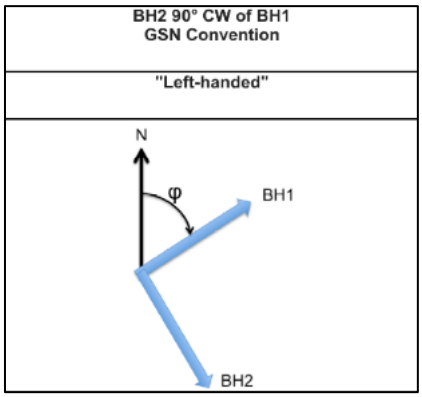
Nomenclature

Pressure channel names

Name hydrophone channels ?DH ("hydrophone").
?Name differential pressure gauge channels "?DF" ("infrasound")?[3]
Name absolute pressure gauge channels "?DO" ("outside")[4] (OO uses "?DO")

Channel Orientation Codes

If the instrument was not oriented along the tradition axes (N-S and E-W), the orientation codes for the horizontal channels should be "1" and "2" according to the geometry specified by the GSN standard shown below. "Azimuth" should be set to 0 for the "1" channel and 90 for the "2" channel. Uncertainties for both should be set to 180.



Site names for repeated deployments

If OBSs are deployed repeatedly at one site (to make a long series), use an incrementing alphanumeric character at the end of the station name, to indicate subsequent deployments (i.e., A01A, then A01B then A01C for subsequent deployments at the same approximate site).

Enter your logger and analog filter information into the NRL

Processing information

OBS data may go through a number of steps before being ready for archival at data centers. These processing steps should be well documented, so that any mistakes can be traced and corrected. The most obvious example is for the timing corrections, but other steps may also be useful. One possibility is to create 'opaque' miniSEED files with this information. Another would be to provide a text file (perhaps structured, such as JSON) with this information. The text or structured file would be more readable, whereas the opaque miniSEED file fits in some data structures (such as SeisComp3 data structure). I think a text file would be easiest for the user to read, but this depends on the data centers having some standard place for these text files (with State of Health data?).
NOTE: CHECK WHAT IRIS USES FOR SUPPORT FILES

[3] Current (IRIS) practice is to name DPG channels ?DH
[4] Corresponds to OO naming convention (verify for IRIS: Cascadia Expt). An alternative would be '?TZ' (tide gauge), though the "pressure" aspect is good to specify

Recommended OBS-specific Software

Standardizing OBS data and metadata storage should also allow efficient validation and correction of OBS-specific problems. Software for at least the following examples should be openly provided:

Clock drift confirmation

Software to calculate clock drift based on drift in hypocenter travel time residuals or noise correlation between instruments

Noise removal

Removal of noise caused by infragravity waves (based on pressure measurements) and dynamic tilt (based on correlation with horizontal channel noise).

Sensor orientation calculation/correction

Determining the horizontal orientation of the instrument using teleseisms and/or local earthquakes. A common method used is the Rayleigh wave polarization method, outlined in the Stachnik et al., 2012 paper with code made available on the OBSIP website (http://www.obsip.org/data/obs-horizontal-orientation/).

Stachnik, J.C., A.F. Sheehan, D.W. Zietlow, Z. Yang, J. Collins, and A. Ferris (2012), Determination of New Zealand Ocean Bottom Seismometer Orientation via Rayleigh-Wave Polarization: Seismol. Res. Lett., 83, 704–712, doi:10.1785/0220110128.

Common pipeline for data preparation

It would be efficient and may avoid incompatibilities if a standard data preparation software was written which input uncorrected miniSEED data and integrated the clock drift correction and any other "standard" processes. The OBS facilities would still be responsible for converting from their proprietary format to miniSEED (with appropriate SEED-based channel names) and to provide clock drift measurements. This would provide the additional advantage that any clock drift corrections found afterwards could be applied almost automatically at the data center level

Active seismic data extractor

A program to input continuous data-center-level data and a shot file (in some standardized format) and output SEG-Y files.

Advantages: Reduce work for OBS facilities (both in writing extraction software and in re-extracting data once clock drift or shot corrections are found). Facilitate data sharing (access on-line at a secured site). Put more OBS data on data-centers (some non-shot parts may be useful for earthquake seismology). Securely archive this data.

OBSIP currently archives active source data in continuous miniSEED and cut SEGY files. The SEGY standard is universal but does not provide instrument responses nor an easy, separate way to modify shot information. PH5 could provide this, but data centers would need to be able to extract miniSEED/etc from it for passive seismic users.