

## 1001 Blockettes Considered Harmful

Dataloggers do not always report time accurately. This is an insidious problem, and dealing with it is excessively complex. The highest priority therefore is to keep timing problems small enough to neglect altogether.

However, when there is a problem, it is important to know it, and also to know how big the problem is. The user needs the information the datalogger knows:

- times the internal clock was synchronized
- the size of any corrections applied
- size of any discontinuous clock corrections.

This information might then be used in processing to ignore some data, or divert it for special handling.

There is little disagreement to this point. It's desirable to have the available data about timing. The blockette 1001 proposal is to incorporate an ad-hoc number related to timing into the actual waveform data, intermixing waveform data and this artificial "clock quality". This is a poorly designed approach to the problem.

What is wrong with this idea?

- First of all, the proposal is not to save the actual data which the datalogger knows, but rather an artificial number. A determined user might attempt to correct the time. The information in the 1001 blockette will not aid this effort.
- The number is to be incorporated into a miniseed blockette, and therefore associated with a particular time range. The particular range of samples that happen to fall into a data block are happenstance, unrelated to the changes that may influence the clock quality number.

In other words, the clock quality number is provided on an uneven time scale of miniseed blocks, which is largely unrelated to the events which dictate the timing accuracy (or the data quality number). For instance, the data from which this ad-hoc number is derived is provided by the Q330 on a 1 sample per second time scale. However, it's put into the miniseed at a rate related to the waveform sample rate and the miniseed block size: anywhere from once a minute to once an hour. The SEED document does not specify how this decimation is to be accomplished; that is, the meaning of clock quality number is unspecified in important ways.

- Waveform data may be (and typically is) resegmented during processing. This is a common and important operation. But it's not obvious how this resegmentation could correctly preserve the clock quality number, short of simply copying the exact block structure of the input. It would be a bad idea to hamstring data processing in this fashion.
- It's not general. Miniseed is not the only data format in the world. Many people use other formats, from various ascii formats to a host of binary formats. None of these other formats will ever support this multiplexed data stream. But the underlying goal should not be the incorporation of 1001 blockettes in all data, but rather to make useful information about the data logger clock available for datastreams.
- It's not extensible. Today, the focus is on the timing, which is certainly important. However, it's not the only important parameter which is not represented in miniseed. For instance, calibration certainly disturbs the waveform and has repercussions in any automated processing, but there is no indication of this in the data stream. mass recenterers cause similar problems. We've recently seen an instance where temperature swings appeared to have a deleterious effect on the waveforms, but temperature is not represented in the miniseed record. What about ambient noise levels? calib is not present in miniseed, but is very important for some analysis.

No doubt, there are many important parameters which could be incorporated into the miniseed, by just adding more and more blockettes. While it may be a good idea to encourage collection of such data, it would be a huge mistake to incorporate it into miniseed as more blockettes.

- It's expensive. This proposal requires rewriting existing interfaces and programs, to support writing, reading and displaying a format specific, artificial quantity with an indistinct relationship to measured quantities. That is, it proposes rewriting working code, inevitably introducing new bugs, to support one data format and one data logger.
- The clock quality code must be derived from the same information separately (and differently) for each waveform channel. It's almost always a bad idea to put the same information in multiple locations.
- Our previous experience with GEOSCOPE multiplexed format -- now largely retired -- should be an indication that multiplexing data may not be the best idea.

## An Alternative

For perspective, consider an alternate approach: save the actual data from which the clock quality is generated as a separate data stream (or streams). In this case, the timing data can be displayed in existing waveform display programs with no change. It can also potentially be used to make corrections. The full resolution and actual time tags of the data are preserved. The data is only saved once, not multiple times (once in each channel of data). It can be easily incorporated into other data formats, even (for example) sac and autoDRM.

## Compare the two approaches:

<b>Feature</b>	<b>Blockette 1001</b>	<b>New Channel</b>
Faithfully represents actual data stream	No	Yes
Can be represented in other dataformats (eg, autoDRM)	No	Yes
May (sometimes) be used to compute corrections	No	Yes
Can be extended to other parameters	No	Yes
Integrates seamlessly with existing software and practice?	No	Yes

## Conclusion

The blockette 1001 proposal attempts to address perceived issues with visibility into timing issues for dataloggers. Unfortunately, it does so in a way which unnecessarily complex and expensive, cannot be generally supported across multiple data formats, and does not provide the information necessary to do anything about the problem.

The seismic community would be better served by a proposal which leveraged existing frameworks, was broadly applicable, and actually provided more useful information.