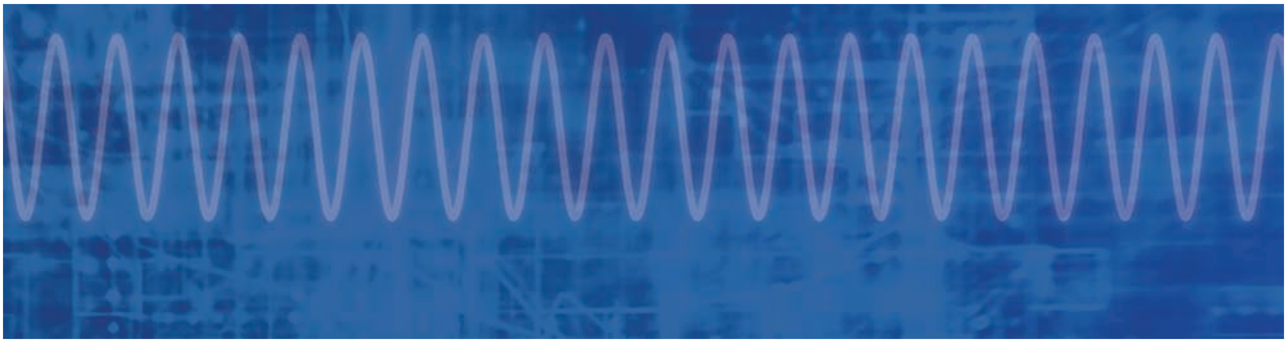
The Institute of Navigation's GNSS SDR Metadata Standard


ION

The past several years has seen a proliferation of software defined radio (SDR) data collection systems and processing platforms designed for GNSS receiver applications or those that support GNSS bands. For post-processing, correctly interpreting the GNSS SDR sampled datasets produced or consumed by these systems has historically been a cumbersome and error-prone process. These systems necessarily produce datasets of various formats, the subtleties of which are often lost in translation when communicating between the producer and consumer of these datasets. This article describes the Institute of Navigation's GNSS SDR metadata standard and associated open-source software project. The initial public comment process for this free and open standard is currently active until December 31, 2017. Readers are encouraged to submit feedback at sdr.ion.org.

JAMES CURRAN
EUROPEAN SPACE AGENCY

MARKEL ARIZABALETA
UNIVERSITÄT DER BUNDESWEHR MÜNCHEN

THOMAS PANY
UNIVERSITÄT DER BUNDESWEHR MÜNCHEN

SANJEEV GUNAWARDENA
AIR FORCE INSTITUTE OF TECHNOLOGY

GNSS SDRs are a rapidly advancing area in GNSS receiver research and design. The last few years have brought tremendous growth in this field. Universities and other research institutions have developed and demonstrated advanced capabilities, particularly with respect to multi-constellation GNSS and GNSS-plus-multi-sensor navigation processing for challenging environments. The recent commercial availability of multi-sensor data collection equipment, development platforms and open-source software projects have catalyzed this rapid pace of innovation. Indeed, SDR will likely become a significant commercial GNSS receiver architecture by the end of this decade due to today's ongoing deployment of multiple global and regional satnav constellations with their diverse signal structures, and rapid advancements in parallel low-power processors and inexpensive sensors.

Non-realtime SDR operational scenarios involve storage and post-processing of samples. These sampled data files can also be used in radio frequency (RF) playback systems for GNSS receiver test and evaluation in the lab. Post-processing and/or playback requires several generic front-end parameters such as RF and intermediate frequency (IF) center frequencies, sample rate, file format, as well as GNSS-specific information such as antenna location and type. We define this information as GNSS SDR metadata. Manual transfer of metadata is the method predominantly used today – a process that is cumbersome and error prone to say the least. No established method exists to exchange this metadata automatically.

During the ION GNSS+ 2013 conference, a group of attendees discussed the need for a formal standard for the exchange of GNSS SDR metadata. This group identified the following benefits of engaging in this activity at the present:

- It identifies and brings the international GNSS SDR community together as a working group. This collaboration is critical in order to get broad acceptance and usage.
- Standardization will help to avoid technology segmentation issues while promoting the pace of innovation by standard practices and compliant tools.
- The formal standard, if widely adopted, will help ensure compatibility and interoperability of future GNSS SDRs. Specifically, front-end agnostic "plug-and-play" SDRs are envisioned. These have the potential for revolutionizing positioning, navigation, and timing (PNT) systems of the future.

Most authors of significant GNSS SDR publications and contributions over the past two decades are ION members and frequent meeting attendees. Hence, we decided to pursue this standard through ION sponsorship. During the January 2014 Council Meeting in San Diego, ION approved the process for establishing a formal standard. The ION GNSS SDR Metadata Working Group (WG) was formed in April 2014. Membership represents academia, industry (including GNSS SDR product vendors as well as traditional GNSS equipment manufacturers), non-profit research entities, and government agencies spanning countries in America, Europe, Asia and Australia.

Justification for GNSS Metadata Standardization

Figure 1 shows the metadata transfer scheme mainly used in today's GNSS SDR systems. The top row depicts data collection system (DCS) *A*, producing an SDR file of format *A*, consumed by SDR processor *A*. This may represent an end-to-end solution provided by a vendor or a system that was initially developed around a specific hardware platform. In any case, assume that the metadata for decoding Format *A* is hard-coded into the processor. Consequently, supporting other file formats involves extensive software revisions on a case-by-case basis. One group may want to share SDR files from DCS *A* with other groups using SDRs *X* and *Y* for research collaboration. This involves conveying the file format and other relevant information accurately to these other groups. Today, this metadata transfer occurs in an ad-hoc way that is prone to interpretation errors.

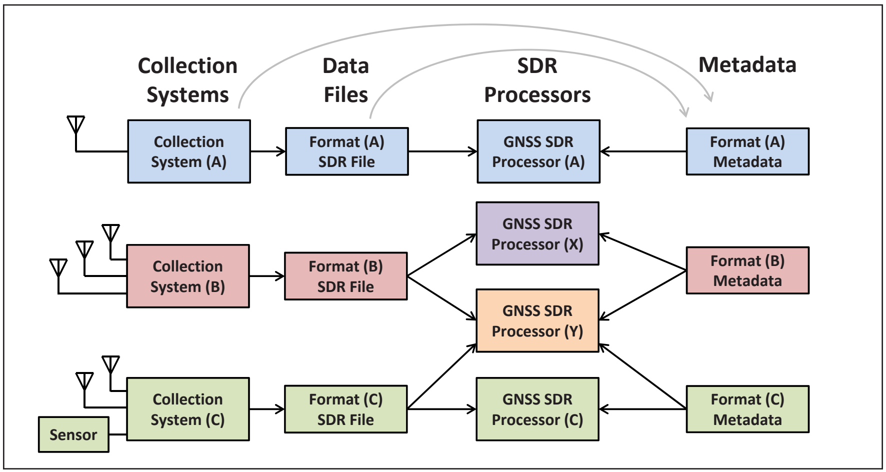


FIGURE 1 Ad-hoc Metadata Exchange

The second row depicts multi-stream DCS *B* that produces files with a more complicated format. SDR processors *X* and *Y* represents more flexible SDRs that are able to support multiple file formats. However, the data/metadata association requires manual intervention.

DCS *C* multiplexes other data (such as sensor data) along with multiple GNSS sample streams in the same file. This type of multiplexed collection has the potential to become more common with emerging SDR-related data streaming standards such as VITA-49 (See T. Cooklev *et alia*, Additional Resources). In this case, custom-designed processor *C* represents an SDR that fully supports multi-sensor integration capabilities. Because DCS *C*'s GNSS stream parameters are open, SDR *Y* is also able to support GNSS-only processing using an ad-hoc metadata transfer scheme. The sensor data parameters in Format *C* may or may not be open.

As clear from Figure 1, today's ad-hoc methods of metadata exchange do not encourage interoperability and instead cultivates potential for technology segmentation (i.e., various groups developing their own stove-piped solutions and technologies).

Figure 2 shows the same systems of Figure 1 that have adopted a metadata standard. As shown, each DCS produces a compliant metadata file along with the SDR file. The metadata file is read-in by the compliant SDR processor to correctly decode and process files seamlessly.

Adoption of a metadata standard benefits DCS developers because their systems become applicable to a much wider group of users. Similarly, an SDR processor's utility is extended when it is capable of supporting many file formats from multiple sources seamlessly. Thus, metadata standardization promotes interoperability of GNSS SDR systems and greatly simplifies the exchange of files between groups.

Metadata standardization also benefits other use cases beyond post-processing GNSS SDRs. For example, consider the use of the metadata specification to synthesize compliant SDR files for use in RF playback systems. Additionally, libraries of compliant SDR file sets containing various real-world scenarios could be used interchangeably in compliant RF playback simulators for repeatable and consistent testing of GNSS receivers.

SDR Data Collection Topologies

The ION Executive Committee stipulated that this standardization activity shall not create an unfair advantage or disadvantage to any entity. Specifically, the standard shall not require any existing system to undergo data format changes to achieve compliance. This "do no harm" stipulation implies that the standard be designed such that it

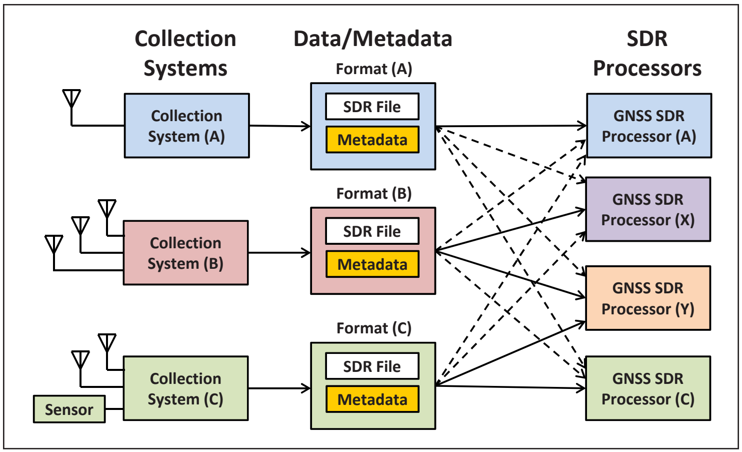


FIGURE 2 Standardized Metadata Exchange

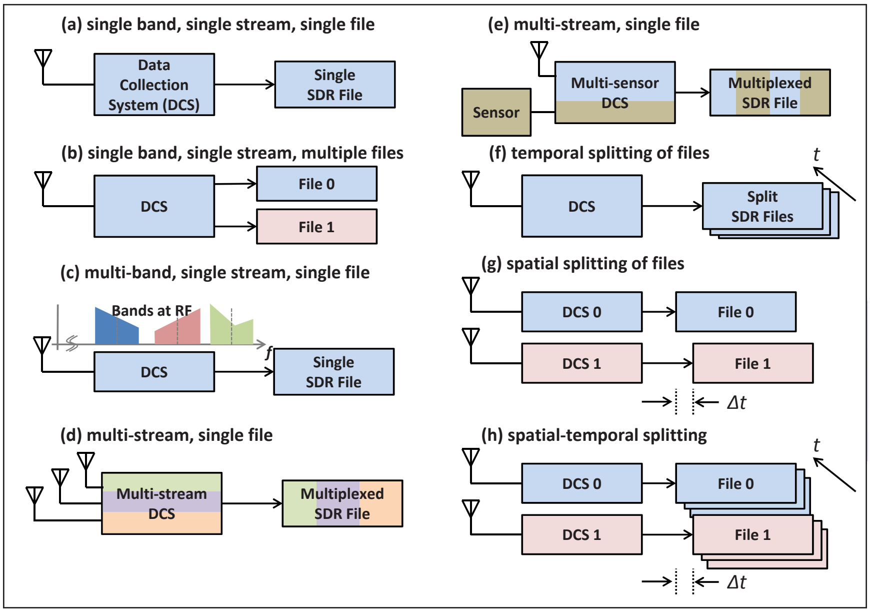

(a) single band, single stream, single file
Data Collection System (DCS)
Single SDR File
(b) single band, single stream, multiple files
DCS
File 0
File 1
(c) multi-band, single stream, single file
Bands at RF
f
DCS
Single SDR File
(d) multi-stream, single file
Multi-stream DCS
Multiplexed SDR File
(e) multi-stream, single file
Sensor
Multi-sensor DCS
Multiplexed SDR File
(f) temporal splitting of files
t
DCS
Split SDR Files
(g) spatial splitting of files
DCS 0
File 0
DCS 1
File 1
Δt
(h) spatial-temporal splitting
t
DCS 0
File 0
DCS 1
File 1
Δt

FIGURE 3 Fundamental GNSS SDR Data Collection Topologies

supports all current and future SDR file formats. This also means that the WG must “get it right the first time” since major revisions to the standard would be undesirable and adverse to the goal of widespread adoption. Hence, we considered the entire space of possible GNSS SDR data collection topologies. Figure 3 illustrates these topologies.

Figure 3.a illustrates the simplest data collection topology that can exist. This is when a single swath of RF spectrum (referenced henceforth as a “band”) is down-converted and sampled to produce a single data stream. The stream, which may be IF sampled (real samples) or baseband sampled (complex samples) is written to disk as a single file.

Figure 3b represents a DCS that writes parts of a single data stream across multiple files. This may be done to reduce the sustained write performance requirement of storage drives (similar to striping mode in RAID systems).

Figure 3.c is similar to Figure 3.a, except that the data stream represents more than one RF band. An example of this topology is a direct RF sampling front-end architecture that intentionally aliases multiple bands to fall next to each other at baseband. Some bands may be spectrally inverted as a result of the digital down-conversion process.

A DCS may produce multiple data streams. Each stream may contain information from one of many antenna elements (as shown in Figure 3.d, where each stream may also encompass multiple bands as in Figure 3.c). Alternatively, a stream may be sampling one of several bands received by a single wideband antenna, where the multiplexed streams written to file represents channelized samples. Combinations of the above are also possible.

Each stream may also be sampled at different rates and bit depths. For example, consider a civilian GPS L1, L2, L5 system. In this case, the L1 and L2 streams may be sampled at rate f_0 and the L5 stream at $10 \cdot f_0$ (since the L5 signal’s null-to-null bandwidth is 10 times wider than L1 C/A and L2C), where f_0 represents the base sample rate. Figure 3.d may also represent how these multiple streams are multiplexed into a single lane of packed binary data and written to a single file.

Similar to Figure 3.d, Figure 3.e illustrates a GNSS data stream multiplexed with other data that is written to a single file. This non-GNSS SDR data may be from additional sensors (as shown), and may be written in a proprietary format that may or may not be known.

The specification of metadata parameters for non-GNSS data is outside the scope of the present standardization effort. However, the standard must support adequate information to skip over such non-GNSS data bytes. Since the metadata schema is extensible, it can cover the description of non-GNSS SDR data as needed by specific users.

It is important to note that although we use the term “GNSS SDR data” in this article to generally refer to the type of sampled data for which metadata parameters are defined in the standard, the samples need not correspond to GNSS frequency bands. For example, frequency bands containing RF signals of opportunity are supported as long as they can be represented by the standard’s metadata parameters.

Due to the typically high data rate of GNSS SDRs, some DCSs write data as temporally split files, as illustrated in Figure 3.f. This allows for efficient data management through multiple smaller files compared to a single large file. The metadata for each file must associate the previous and next files in order to represent the sequence. Note that the DCS block shown in Figure 3.f could represent any of those described in topologies a, c, d, and e of Figure 3.

Figure 3.g illustrates spatial splitting of files. Here, the binary data lanes from two or more DCSs are written to separate files. These files may be written by different computer systems. In this case, a non-zero inter-system timing offset, Δt, may exist and must be supported in the standard. Since Δt may or may not be known at time of collection, it represents an example metadata parameter that may be back-annotated into the metadata file after an initial pass of SDR data processing.

Multi-lane DCSs that write each lane to a separate file may also implement temporal file splitting according to 3.f. We call this "spatial-temporal splitting", and is illustrated in Figure 3.h.

Since multi-stream-multi-file data collection topologies (i.e., (g) and (h) in Figure 3) produce multiple data files applicable to a given time interval, the SDR processor must be "introduced" to the full or partial set of lanes associated with the topology. This is covered by lane selection parameters in the standard.

Metadata Parameters

The metadata standard enables a user to specify a wide range of properties of the binary data file. Some of these properties relate to the data collection scenario itself, such as the time and location, and the type of scenario. Other properties of the dataset relate to the particular RF data that is captured as IF samples, including center frequency and bandwidth. Yet other details such as the IF digitization configuration and data packing can be specified. These are briefly summarized below. Some of the parameters are free-form text, others are integer or floating point variables, and others are enumerations. The primary classes are shown in **Figure 4**.

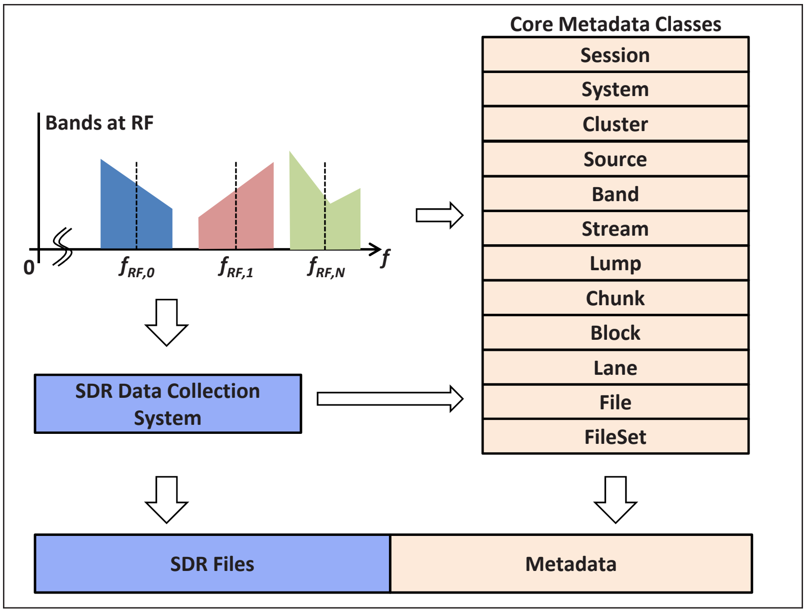


FIGURE 4 Standardized Metadata Exchange

The metadata "session" class holds details such as the time, the position, the measurement campaign, and the type of scenario. The "file" class contains a path to the corresponding binary data file, time-stamp info, and a reference to the contained "lane" (described below). A "system" class contains details of the data recording equipment, such as the equipment name, the reference clock frequency, and the antenna details/specifications. The metadata standard allows for the specification of a variety of RF bands, each "band" class includes details of the band's center frequency, intermediate frequency, bandwidth and group-delay bias.

The actual format of the binary data is detailed in the "lane" class, which allows for the specification of a hierarchy of binary data packing patterns, entitled "blocks", "chunks" and "lumps". These classes specify the arrangement of the packed IF data and optional additional data such as sensor measurements or configuration information, and further specify the arrangement of data relative to the standard word-sizes (arrays of byte, short integers, integers, long integers, etc.). The arrangement of the raw IF samples within these blocks is specified by the "stream" class. This class specifies the associated RF band, the sample rate, the quantization, the sample format (real/complex), the encoding of the binary data and the arrangement/alignment of the samples. Together these classes contain sufficient information to unambiguously interpret the packed binary data.

Normative Reference Software

The primary responsibilities of the Normative Reference Implementation Subcommittee are to reduce to practice the conceptual design developed through consensus by the WG into an appropriate set of schema (according to industry best practices) and to develop a compliant software library implementation. The WG was fortunate to receive voluntary participation for this task from individuals representing established commercial GNSS SDR vendors.

The WG decided to work on a publicly available reference software library to be released with the standard. The goal of this effort is to promote early and widespread adoption of the standard by making it easy for vendors and researchers to integrate standard-compliance by integrating the library into their existing software. The scope of this effort is twofold, to develop a metadata interpreter, and to develop a binary data converter using this metadata interpreter.

The first contribution was a library that would produce standard-compliant metadata files from an appropriately-populated data structure, a library capable of reading the content of such files back to a data structure. The second contribution was in the form of a library capable of parsing and converting binary IF data based on an associated metadata description. When combined, it is hoped that these two libraries offer sufficient functionality to enable adoption of the metadata standard in SDRs, or to serve as a benchmark against which implementations of the standard can be validated.

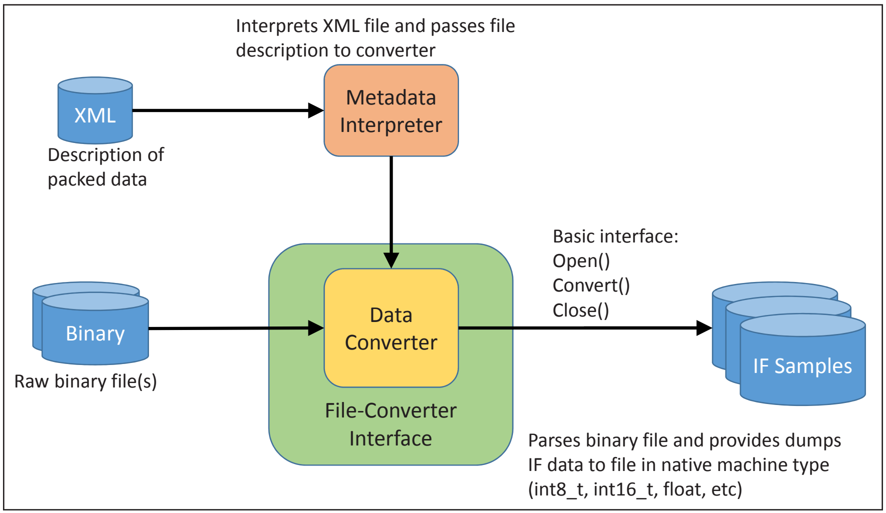

Interprets XML file and passes file description to converter
XML
Description of packed data
Metadata Interpreter
Basic interface:
Open()
Convert()
Close()
Binary
Raw binary file(s)
Data Converter
File-Converter Interface
IF Samples
Parses binary file and provides dumps IF data to file in native machine type (int8_t, int16_t, float, etc)

FIGURE 5 Example of the Normative Reference Software configured as a file converter

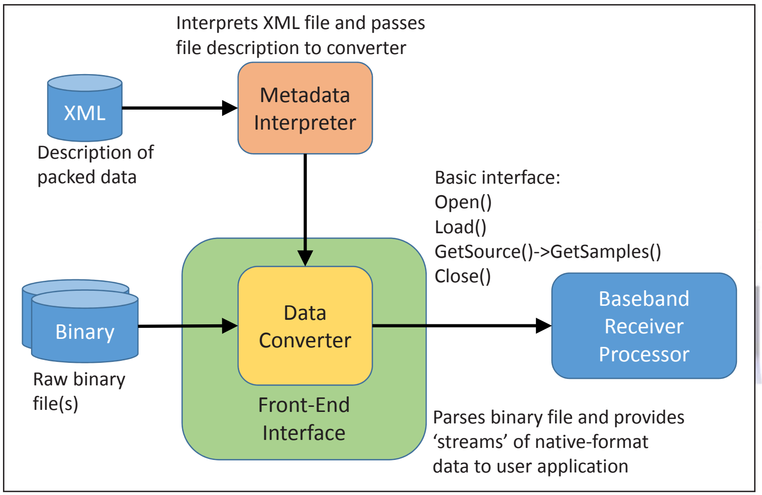

Interprets XML file and passes file description to converter
XML
Description of packed data
Metadata Interpreter
Basic interface:
Open()
Load()
GetSource()->GetSamples()
Close()
Binary
Raw binary file(s)
Data Converter
Front-End Interface
Baseband Receiver Processor
Parses binary file and provides 'streams' of native-format data to user application

FIGURE 6 Example of the Normative Reference Software configured as a software GNSS receiver front-end

The software is written in C++ and managed using CMake. It has been divided into two libraries, "apilib" implementing the Metadata Interpreter and "converterlib" implementing the binary data conversion, and is accompanied by a selection of utilities, including a data-converter application and a simple test application. The software is available on the Institute of Navigation's GitHub account at the following link: < https://github.com/IonMetadataWorkingGroup/GNSS-Metadata-Standard>.

The Metadata Interpreter library includes a reader functionality that can parse a metadata file and populate a corresponding metadata object that can then be queried through a selection of member-function calls. Similarly, a metadata object can be instantiated and configured through a selection of member functions and, subsequently, it can be instructed to write a corresponding metadata file.

The converter library can be used for parsing binary data files and interpreting the data according to a metadata file. The basic converter can be adapted to support processing of the converted data streams, and two such adaptations have been implanted in the reference software. The first functionality is depicted in **Figure 5**, where the data-converter is embedded in a file-converter. The file-converter configures the embedded data-converter using a Metadata Interpreter object, and is capable of parsing a packed binary input file and producing one file per IF data stream in a user-specified data type (int8, int16, float, double, etc.).

A second functionality has been implemented by embedding the data converter in a "front-end", as depicted in **Figure 6**. This front-end offers a means of loading short portions of the binary data file and converting them to a user-specified data type (int8, int16, float, double, etc.) while handling details such as sample alignment, when different streams are sampled at different rates.

The software suite includes a selection of example binary datasets and associated metadata files along with a simple MATLAB/Octave script to test the build against reference datasets. To date five different file formats have been included in the repository including a wide range of front-end configurations and data packing variations.

A number of working group members volunteered to perform "blind testing" of SDR data files against draft specifications. This involves exchanging SDR data files and associated metadata specifications among parties and verifying that the files can be fully decoded without additional information. Working group members participating in this activity will comprise the Compliance Verification Subcommittee.

SDR Data Repository

As with most standards and programming projects, they are best understood through good examples. Thus, the page: http://sdr.ion.org/api-sample-data.html has been created and contains several binary sample files together with metadata files. All file sets have been tested to follow the standard and to be readable by the normative reference software. The binary files typically contain samples over a duration of more than 60 seconds and position fixes have been obtained by at least one software receiver implementation.

Further examples will be added, not only emphasizing new data recording systems but also illustrating different SDR applications such as antenna arrays, reflectometry or ionospheric scintillation analysis. Further, data from GNSS seen only at certain locations of the

Earth (e.g., Quasi-Zenith Satellite System [QZSS]) are being considered.

Organizations are encouraged to contact the working group if they have files with formats that are currently not adequately represented in the repository or represent use cases so far not considered. The respective point of contact is given on the web page. The working group will then take care of creating a metadata file (if not yet available), check the correctness and organize the upload.

Additional Resources

[1] Cooklev, T., R. Normoyle, and D. Clendenen, "The Vita49 Analog RF-Digital Interface," *IEEE Circuits and Systems Magazine*, Q4 2012.

[2] Favenza, A., and N. Linty, "Exploiting Standardized Metadata for GNSS SDR Remote Processing: a Case Study", ION GNSS+ 2016, Portland, Oregon (USA), 09/2016.

[3] Lucas-Sabola, V., G. Seco-Granados, J. A. López-Salcedo, J. A. García-Molina, M. Crisci, "Cloud GNSS receivers: New advanced applications made possible", *Proc. Int. Conf. Localization GNSS (ICL-GNSS)*, pp. 1-6, 2016.

[4] Lucas-Sabola, V., Seco-Granados, G., López-Salcedo, J.A., García-Molina, J.A., Crisci, M., "Demonstration of Cloud GNSS Signal Processing," *Proceedings of the 29th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2016),* Portland, Oregon, September 2016, pp. 34-43.

Authors

James T. Curran (jamestcurran@ieee.org) received a B.E. in electrical & electronic Engineering and a Ph.D. in telecommunications from the Department of Electrical Engineering, University College Cork, Ireland. He worked as a senior research engineer with the PLAN Group in the University of Calgary, Canada, and as a grant-holder at the Joint Research Centre (JRC) of the European Commission (EC), Italy. James is currently a radionavigation engineer at the European Space Agency in the Netherlands. His main research interests are signal processing, information theory, cryptography, and software defined radio.

Markel Arizabaleta has a M.Sc. from the University of the Basque Country (UPV-EHU) in Telecommunication Engineering. He studied the last year in Tampere University of Technology (TUT) where he worked on joint 5G mobile communication and positioning systems and joined the Universität der Bundeswehr München in 2017 where he is involved in signal processing for satellite navigation systems.

Thomas Pany is with the Universität der Bundeswehr München at the faculty of aerospace engineering where he teaches satellite navigation. His research includes all aspects of navigation ranging from deep space navigation over new algorithms and assembly code optimization. Currently he focuses on GNSS signal processing for Galileo second generation, GNSS receiver design and GNSS/INS/LiDAR/camera fusion. To support these activities, Pany is developing a modular GNSS test bed for advanced navigation research. Previously he worked for IFEN GmbH and IGASPIN GmbH and is the architect of the ipexSR and SX3 software receiver. Pany has around 200 publications including patents and one monography.

Sanjeev Gunawardena is a Research Assistant Professor of Electrical Engineering with the Autonomy & Navigation Technology (ANT) Center at the Air Force Institute of Technology (AFIT). His research interests include RF design, digital systems design, high performance computing, software defined radio, and all aspects of GNSS receivers and associated signal processing.

Make High-Precision Applications Affordable and Easy
Tersus GNSS L1/L2 RTK Boards
CM-level Positioning Accuracy
• GPS, GLONASS, BeiDou
• Dual Frequency
• Ideal for UAVs, Surveying, Precision Agriculture…
TERSUS
www.tersus-gnss.com
info@tersus-gnss.com