Initial Report of Activities of the GNSS SDR Metadata Standard Working Group

Session C4: Software Receivers
September 11, 2014
Tampa, FL
Background

• Proliferation of GNSS SDR technology in the past 5-10 years
  – Low-cost front-end hardware and data collection systems
  – Maturing GNSS SDR processors, receivers and software frameworks
Background

• Today: no established standard to convey GNSS SDR metadata
  – Existing metadata standards not well suited for needs of GNSS SDR and PNT community
    • VITA-49 (packet based, sample encoding already specified)
    • GNU Radio metadata standard (binary header based)

• ION SDR Metadata Standard
  – Objective: Interoperability between GNSS SDR data collection systems and processors
  – ‘Do no harm’
Actions to Date

• Jan, 2014
  – ION Council approved formation of working group

• Feb, 2014 – Selection of Co-Chairs:
  – Dr. Sanjeev Gunawardena
  – Dr. Thomas Pany

• Apr, 2014
  – ION Executive Council reviewed and approved invite list and documented process to follow
  – Approx. 95 Invitations sent to representative international cross section of GNSS SDR subject matter experts from academia, industry and government
  – Invite material included ION legal reviewed forms to be returned with acceptance
    • Conflict of Interest Disclosure Statement
    • Request for Patents and Participant’s Position Regarding Licensing of Essential Patent Claims
  – Returned forms reviewed by ION Executive Director
Actions to Date (contd.)

• May, 2014
  – Established Working Group (49 members) and ION-administrated online discussion group
  – Posted initial draft (developed by co-chairs) for WG feedback
  – Online WG discussions

• Aug, 2014
  – WG feedback incorporated into second draft and posted for discussion

• Sep, 2014
  – Discuss and finalize Initial Report and standard First Draft at first in-person WG meeting on Tuesday Sept 9 from 15:10 to 17:10
  – Initial Report Presentation at Software Radio Session (C4) on Thursday, Sep. 11 at 17:08
## Working Group Members

<table>
<thead>
<tr>
<th>Name</th>
<th>Company/Affiliation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Akos, Dennis M.</td>
<td>University of Colorado</td>
</tr>
<tr>
<td>Al-Masyabi, Walid</td>
<td>Raytheon</td>
</tr>
<tr>
<td>Bavaro, Michele</td>
<td>One Talent GNSS</td>
</tr>
<tr>
<td>Belabbas, Boubeker</td>
<td>DLR</td>
</tr>
<tr>
<td>Braasch, Michael</td>
<td>Ohio University</td>
</tr>
<tr>
<td>Chansarkar, Mangesh</td>
<td>CSR</td>
</tr>
<tr>
<td>Cosgrove, Mathew</td>
<td>Northrop Grumman NSD</td>
</tr>
<tr>
<td>Crampton, Paul G.</td>
<td>Spirent Federal Systems</td>
</tr>
<tr>
<td>Dovis, Fabio</td>
<td>Politecnico di Torino</td>
</tr>
<tr>
<td>Eissfeller, Bernd</td>
<td>University FAF</td>
</tr>
<tr>
<td>Fernández Hernández, Ignacio</td>
<td>Galileo Supervisory Agency</td>
</tr>
<tr>
<td>Fernández-Prades, Carles</td>
<td>Centre Tecnològic de Telecomunicaciones de Catalunya (CTTC)</td>
</tr>
<tr>
<td>Gavrilov, Artyom</td>
<td>GNSS-SDR</td>
</tr>
<tr>
<td>Glennon, Eamonn</td>
<td>University of New South Wales</td>
</tr>
<tr>
<td>Goodrich, Brian</td>
<td>NavCom</td>
</tr>
<tr>
<td>Gunawardena, Sanjeev</td>
<td>Air Force Institute of Technology</td>
</tr>
<tr>
<td>Kou, Yanhong</td>
<td>Beihang University</td>
</tr>
<tr>
<td>Kubo, Nobuaki</td>
<td>Tokyo University of Marine Science and Technology</td>
</tr>
<tr>
<td>Langer, Markus</td>
<td>Karlsruhe Institute of Technology (KIT)</td>
</tr>
<tr>
<td>Ledvina, Brent</td>
<td>Coherent Navigation</td>
</tr>
<tr>
<td>Little, Jon C.</td>
<td>Applied Research Laboratories of the University of Texas at Austin</td>
</tr>
<tr>
<td>Lohan, Elena-Simona</td>
<td>Tampere University of Technology</td>
</tr>
<tr>
<td>López-Almansa, José María</td>
<td>GMV</td>
</tr>
<tr>
<td>Lopez-Risueño, Gustavo</td>
<td>European Space Agency</td>
</tr>
<tr>
<td>Mathews, Michael B.</td>
<td>Loctronix</td>
</tr>
<tr>
<td>Morton, Yu (Jade)</td>
<td>Colorado State University</td>
</tr>
<tr>
<td>O’Brien, Andrew J.</td>
<td>Ohio State University</td>
</tr>
<tr>
<td>Pany, Thomas</td>
<td>Ifen GmbH</td>
</tr>
<tr>
<td>Parsons, Bryan Masamitsu</td>
<td>Applied Research Laboratories of the University of Texas at Austin</td>
</tr>
<tr>
<td>Pelosi, Lou</td>
<td>Cast Navigation</td>
</tr>
<tr>
<td>Petovello, Mark</td>
<td>University of Calgary</td>
</tr>
<tr>
<td>Pinchin, James</td>
<td>University of Nottingham</td>
</tr>
<tr>
<td>Psiaki, Mark</td>
<td>Cornell University</td>
</tr>
<tr>
<td>Rudra, Angsuman</td>
<td>D-TA Systems</td>
</tr>
<tr>
<td>Rügamer, Alexander</td>
<td>Fraunhofer IIS</td>
</tr>
<tr>
<td>Schipper, Brian</td>
<td>Honeywell</td>
</tr>
<tr>
<td>Schleppe, John B.</td>
<td>NovAtel</td>
</tr>
<tr>
<td>Scott, Logan</td>
<td>LS Consulting</td>
</tr>
<tr>
<td>Shivaramaiah, Nagaraj</td>
<td>GNSS Labs</td>
</tr>
<tr>
<td>Soloviev, Andrey</td>
<td>Qunav</td>
</tr>
<tr>
<td>Suzuki, Taro</td>
<td>Tokyo University of Marine Science and Technology</td>
</tr>
<tr>
<td>Tkatch, Alex</td>
<td>Rohde &amp; Schwarz USA Inc.</td>
</tr>
<tr>
<td>Vinande, Eric</td>
<td>AFRL Sensors Directorate</td>
</tr>
<tr>
<td>Wesson, Kyle</td>
<td>University of Texas at Austin/Zeta Associates</td>
</tr>
<tr>
<td>WON, Jong-Hoon</td>
<td>ISTA at University FAF Munich</td>
</tr>
<tr>
<td>Yang, Ning</td>
<td>Draper Laboratory</td>
</tr>
<tr>
<td>Yao, Zheng</td>
<td>Tsinghua University</td>
</tr>
<tr>
<td>Yu, Jim</td>
<td>Trimble</td>
</tr>
<tr>
<td>Yu-Hsuan Chen</td>
<td>Stanford University</td>
</tr>
</tbody>
</table>
Summary of 9/9/14 WG Meeting

• Attendance: 30+, mostly WG members
• Motions passed, items agreed upon:
  • XML as metadata description format
  • ‘SDRX’ as metadata file extension
  • Documentation subcommittee
  • ‘Metadata codec’ reference implementation
  • Alpha testing
• 2\textsuperscript{nd} draft standard accepted in principal with proposed revisions
• Several action items
Next Steps

• **Oct 13** : Initial Draft Standard

• **Oct 13, 2014 to Jan 12, 2015** : 90-day Open Public Comment Period
  - ION.ORG home page notice during open comment period
  - Published on ION.ORG: Initial Report, Initial Draft, online comment form (similar to ION abstract portal)
  - WG members encouraged to initiate informal invitations to GNSS/PNT community for comment
  - WG members will review and discuss public comments as and when received

• **Jan 26, 2015** – In-person WG meeting at ION ITM 2015
  - Proposed Draft Standard activities
  - Follow-on report at ION ITM
  - Some adoption of draft standard (reference implementation subcommittee)
The Problem

- Some front-end/DCS and SDR processors are ‘married’ to one another
- Ad hoc metadata exchange – prone to human error
- Does not promote interoperability
- Does not promote data/resource sharing and re-use
Proposed Solution: Metadata Standardization

- Unambiguous transfer of all essential SDR metadata
- Standardization encourages vendors to support major formats
- Spurs community to develop open-source software handlers and plug-ins
- Promotes interoperability
- Promotes data portability, resource sharing and re-use
Challenges

• Many SDR Data Collection Topologies
  – We’re not defining SDR data standard
  – Need to get it ‘mostly correct’ on first try
    • Minor revisions okay
    • Maintain backwards compatibility between revisions

• Ensuring Widespread Adoption
  – Early adoption by academia & research labs is key
  – Open-source ‘metadata codecs’ and format handlers

• Vendor Support
  – Actively pursue vendor participation in WG
  – Report/demonstrate success of early-adoption efforts
  – Work through any ‘proprietary format’ issues
SDR Data Collection Topologies

Single band, single-stream, single file

Data Collection System (DCS) → Single SDR File

Multi-band, single-stream, single file

DCS → Single SDR File

Bands at RF

Spatial splitting of files

DCS 0 → File 0

DCS 1 → File 1

Δt

Spatial-temporal splitting

DCS 0 → File 0

DCS 1 → File 1

Δt

Temporal splitting of files

DCS → Split SDR Files

Single band, single-stream, single file

Data Collection System (DCS) → Single SDR File

Multi-stream, single file

Multi-stream DCS → Multiplexed SDR File

Multi-stream, single file

Multi-sensor DCS → Multiplexed SDR File

IMU
Metadata Parameters

To support various topologies, data formats and their combinations, the SDR metadata standard defines a set of parameters to fully describe SDR file(s). We refer to these as metadata parameters. The figure below depicts the hierarchy of metadata parameters.

The following slides define each of these Metadata Parameters.
eXtensible Markup Language (XML) as Metadata Format

- Free and open standard
- Wide industry adoption
- API support in all OS and app development frameworks (C/C++, python, MATLAB, ...)
- Lightweight, text-based format
- Human readable/editable
- Extensibility (supports proprietary metadata)
- Supports back-annotation
System Parameters

- Specifies top-level parameters related to data collection campaign and equipment

Table 1. System Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Enumeration</th>
<th>Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time zone</td>
<td>TIMEZONE</td>
<td>TBD</td>
<td>string (3 letters?)</td>
<td></td>
</tr>
<tr>
<td>Lowest common factor sample rate, $f_s$</td>
<td>FSBASE</td>
<td></td>
<td>UINT32.UINT32 (N/D)</td>
<td>samples/second</td>
</tr>
<tr>
<td>Information relating to data collection campaign</td>
<td>CAMPAIGN</td>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Scenario contained in data</td>
<td>SCENARIO</td>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Information on data collection equipment and/or configuration</td>
<td>EQUIPMENT</td>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Antenna: make, model, type, etc.</td>
<td>ANTIENA</td>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>approximate (antenna) location of data collection. Lat, Lon, Height coordinates</td>
<td>LOCATION</td>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Point-of-contact for data collection (person’s name)</td>
<td>POC</td>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Contact info for POC. E.g. email</td>
<td>CONTACT</td>
<td></td>
<td>string</td>
<td></td>
</tr>
</tbody>
</table>
System Parameters: XML

<!-- System Parameters -->
<TIMEZONE>UTC</TIMEZONE>
<FSBASE>56320000.1</FSBASE>
<CAMPAIGN>Collection Campaign</CAMPAIGN>
<SCENARIO>Specific Scenario</SCENARIO>
<EQUIPMENT>Equipment used: Make/Mode/Serial</EQUIPMENT>
<ANTENNA>Antenna Make/Model</ANTENNA>
<LOCATION>Location Coordinates</LOCATION>
<POC>Entity or person that collected data</POC>
<CONTACT>Contact information</CONTACT>
**Band Definition and Parameters**

- A band is defined as a finite span of radio frequency (RF) spectrum.
- Band parameters include information on frequency translation to baseband.

![Diagram of Bands at RF and Baseband](image)

### Table 1. Band Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Enumeration</th>
<th>Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of bands</td>
<td>NUMBANDS</td>
<td>UINT16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>containing-stream index</td>
<td>STRIDX</td>
<td>UINT8</td>
<td></td>
<td>zero-referenced index</td>
</tr>
<tr>
<td>unique identifier</td>
<td>NAME</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF center frequency</td>
<td>FRF</td>
<td>UINT32.UINT32 (N/D)</td>
<td>UINT32.UINT32 (N/D)</td>
<td>cycles/second</td>
</tr>
<tr>
<td>translated center frequency</td>
<td>FIF</td>
<td>UINT32.UINT32 (N/D)</td>
<td>UINT32.UINT32 (N/D)</td>
<td>cycles/second</td>
</tr>
<tr>
<td><strong>spectral inversion</strong> flag</td>
<td>INV</td>
<td>0: non-inverted 1: inverted</td>
<td>UINT8</td>
<td></td>
</tr>
<tr>
<td>approximate double-sided half-power bandwidth</td>
<td>BW3</td>
<td>double</td>
<td>cycles/second</td>
<td></td>
</tr>
</tbody>
</table>
<NUMBANDS>4</NUMBANDS>
<BAND STRIDX="0" NAME="L1 ANT1" FRF="1575420000.1" FIF="13680000.1"
INV="0" BW3="20000000.1">
</BAND>
<BAND STRIDX="1" NAME="L2 ANT1" FRF="1227600000.1" FIF="13680000.1"
INV="0" BW3="24000000.1">
</BAND>
<BAND STRIDX="2" NAME="G1 ANT1" FRF="1602000000.1" FIF="13680000.1"
INV="0" BW3="26000000.1">
</BAND>
<BAND STRIDX="3" NAME="G2 ANT1" FRF="1246000000.1" FIF="13680000.1"
INV="0" BW3="26000000.1">
</BAND>
Stream

- A stream is defined as a series of binary values that are derived from samples produced by a digitizing radio frequency front-end or equivalent source.
- Stream sample rate specified as integer multiple ($SF$) of the system base sample rate $f_s$
- Stream data contains information from one or more bands
- Front-end samples are conditioned and packed into a stream according to the following stream parameters:
  - Baseband or IF sampling (including inversion of numerical values)
  - Sample quantization (for baseband sampling, I & Q samples treated as two distinct values with same quantization. i.e. two values for each sample)
  - Packed bits per sample
  - Sample encoding scheme
  - Packed sample alignment

![Diagram of Stream Process]

$\text{Sample conditioning/packing}$

$S_k \quad S_{k+1} \quad S_{k+2}$

Stream 0

---

ION GNSS SDR Metadata Working Group
## Stream Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Enumeration</th>
<th>Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of streams</td>
<td>NUMSTR</td>
<td></td>
<td>UINT16</td>
<td></td>
</tr>
<tr>
<td>unique identifier</td>
<td>NAME</td>
<td></td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>sample rate factor</td>
<td>SF</td>
<td></td>
<td>UINT8</td>
<td>integer scale factor</td>
</tr>
<tr>
<td>sampling method and representation</td>
<td>IF, IF’, IQ, IQ’, I’Q, I’Q, QI, QI’, Q’I, Q’I’ (where ‘ signifies inversion)</td>
<td>string</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sample quantization</td>
<td>QTZ</td>
<td></td>
<td>UINT8</td>
<td>bits/sample</td>
</tr>
<tr>
<td>Packed bits/sample</td>
<td>BTP</td>
<td></td>
<td>UINT8</td>
<td>bits/sample</td>
</tr>
<tr>
<td>Sample packing alignment</td>
<td>ALN</td>
<td>‘L’ – left aligned ‘R’ – right aligned ‘N’ – not applicable</td>
<td>string</td>
<td></td>
</tr>
<tr>
<td>Sample encoding</td>
<td>ENC</td>
<td>SIGN, SM, INT, BIN, FP</td>
<td>string</td>
<td></td>
</tr>
</tbody>
</table>
### Sample Encoding Schemes

<table>
<thead>
<tr>
<th>Bits (QTZ)</th>
<th>Encoding</th>
<th>Set</th>
<th>Range Min</th>
<th>Range Max</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>sign</td>
<td>{-1, +1}</td>
<td>-1</td>
<td>+1</td>
<td>SIGN</td>
</tr>
<tr>
<td>2</td>
<td>sign-magnitude</td>
<td>{-3, -1, +1, +3}</td>
<td>-3</td>
<td>+3</td>
<td>SM2</td>
</tr>
<tr>
<td></td>
<td>signed integer</td>
<td>{-2, -1, 0, 1}</td>
<td>-2</td>
<td>+1</td>
<td>INT2</td>
</tr>
<tr>
<td></td>
<td>offset binary</td>
<td>{-2, -1, 0, 1}</td>
<td>-2</td>
<td>+1</td>
<td>BIN2</td>
</tr>
<tr>
<td>3</td>
<td>sign-magnitude</td>
<td>{-7, -5, -3, -1, +1, +3, +5, +7}</td>
<td>-7</td>
<td>+7</td>
<td>SM3</td>
</tr>
<tr>
<td></td>
<td>signed integer</td>
<td>{-4, -3, -2, -1, 0, 1, 2, 3}</td>
<td>-4</td>
<td>+3</td>
<td>INT3</td>
</tr>
<tr>
<td></td>
<td>offset binary</td>
<td>{-4, -3, -2, -1, 0, 1, 2, 3}</td>
<td>-4</td>
<td>+3</td>
<td>BIN3</td>
</tr>
<tr>
<td>4</td>
<td>sign-magnitude</td>
<td>{-8, -7, ..., -1, +1, ..., +8}</td>
<td>-8</td>
<td>+8</td>
<td>SM4</td>
</tr>
<tr>
<td></td>
<td>signed integer</td>
<td>{-8, ..., 0, ..., +7}</td>
<td>-8</td>
<td>+7</td>
<td>INT4</td>
</tr>
<tr>
<td></td>
<td>offset binary</td>
<td>{-8, ..., 0, ..., +7}</td>
<td>-8</td>
<td>+7</td>
<td>BIN4</td>
</tr>
<tr>
<td>8</td>
<td>sign-magnitude</td>
<td>{-128, -127, ..., +127, +128}</td>
<td>-127</td>
<td>+128</td>
<td>SM8</td>
</tr>
<tr>
<td></td>
<td>signed integer</td>
<td>{-128, ..., 0, ..., 127}</td>
<td>-128</td>
<td>+127</td>
<td>INT8</td>
</tr>
<tr>
<td></td>
<td>offset binary</td>
<td>{-128, ..., 0, ..., 127}</td>
<td>-128</td>
<td>+127</td>
<td>BIN8</td>
</tr>
<tr>
<td>16</td>
<td>sign-magnitude</td>
<td>{-2^{15}, ..., -1, +1, ..., +2^{15}}</td>
<td>-2^{15}</td>
<td>+2^{15}</td>
<td>SM16</td>
</tr>
<tr>
<td></td>
<td>signed integer</td>
<td>{-2^{15}, ..., 0, ..., 2^{15}-1}</td>
<td>-2^{15}</td>
<td>+2^{15}-1</td>
<td>INT16</td>
</tr>
<tr>
<td></td>
<td>offset binary</td>
<td>{-2^{15}, ..., 0, ..., 2^{15}-1}</td>
<td>-2^{15}</td>
<td>+2^{15}-1</td>
<td>BIN16</td>
</tr>
<tr>
<td></td>
<td>floating point</td>
<td>IEEE 754-2008, FP16</td>
<td></td>
<td></td>
<td>FP16</td>
</tr>
<tr>
<td>32</td>
<td>sign-magnitude</td>
<td>{-2^{31}, ..., -1, +1, ..., +2^{31}}</td>
<td>-2^{31}</td>
<td>+2^{31}</td>
<td>SM32</td>
</tr>
<tr>
<td></td>
<td>signed integer</td>
<td>{-2^{31}, ..., 0, ..., 2^{31}-1}</td>
<td>-2^{31}</td>
<td>+2^{31}-1</td>
<td>INT32</td>
</tr>
<tr>
<td></td>
<td>offset binary</td>
<td>{-2^{31}, ..., 0, ..., 2^{31}-1}</td>
<td>-2^{31}</td>
<td>+2^{31}-1</td>
<td>BIN32</td>
</tr>
<tr>
<td></td>
<td>floating point</td>
<td>IEEE 754-2008, FP32</td>
<td></td>
<td></td>
<td>FP32</td>
</tr>
<tr>
<td>64</td>
<td>sign-magnitude</td>
<td>{-2^{63}, ..., -1, +1, ..., +2^{63}}</td>
<td>-2^{63}</td>
<td>+2^{63}</td>
<td>SM64</td>
</tr>
<tr>
<td></td>
<td>signed integer</td>
<td>{-2^{63}, ..., 0, ..., 2^{63}-1}</td>
<td>-2^{63}</td>
<td>+2^{63}-1</td>
<td>INT64</td>
</tr>
<tr>
<td></td>
<td>offset binary</td>
<td>{-2^{63}, ..., 0, ..., 2^{63}-1}</td>
<td>-2^{63}</td>
<td>+2^{63}-1</td>
<td>BIN64</td>
</tr>
<tr>
<td></td>
<td>floating point</td>
<td>IEEE 754-2008, FP64</td>
<td></td>
<td></td>
<td>FP64</td>
</tr>
</tbody>
</table>

Other QTZs and encodings exist. WG should determine if we need to specify all of them.
Stream Parameters (contd.)

- QTZ, BTP, and ALN specify how extra bits are discarded in the decoding process.

- ALN is ‘N’ when QTZ equals BTP (i.e. no extra bits to consider in packed format)

- When BTP is greater than QTZ, **discard extra bits when decoding** (i.e. do not assume sign extension or zero padding)

- QTZ, BTP and ALN parameters only specify how quantized samples of a specified encoding are packed. They do not specify which bits are MSB/LSB. This information is specified in chunk parameters.

**XML:**

```xml
<!-- Stream Parameters -->
[NUMSTR]>4</NUMSTR>
<STREAM NAME="S0" SF="1" QTZ="1" BTP="1" ALN="N" ENC="SIGN"/></STREAM>
<STREAM NAME="S1" SF="1" QTZ="1" BTP="1" ALN="N" ENC="SIGN"/></STREAM>
<STREAM NAME="S2" SF="1" QTZ="1" BTP="1" ALN="N" ENC="SIGN"/></STREAM>
<STREAM NAME="S3" SF="1" QTZ="1" BTP="1" ALN="N" ENC="SIGN"/></STREAM>
```
Samples from multiple coherently-sampled streams may be multiplexed in time to form a single ‘collection’ of sequential samples. This requires parameters to describe how streams are multiplexed. The term ‘chunk’ is used later to describe a segment of data for which an encoding/decoding scheme is specified. Hence, the term ‘lump’ is used to describe the collection of all binary data that gets generated during a single $1/f_s$ sampling epoch.

A lump is defined as the ordered containment of all samples occurring within the interval $t_s=1/f_s$.
### Lump Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Enumeration</th>
<th>Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of multiplexed streams</td>
<td>NMXSTR</td>
<td>-</td>
<td>UINT8</td>
<td>Streams</td>
</tr>
<tr>
<td>stream indices (in order of appearance)</td>
<td>STRIDX</td>
<td>-</td>
<td>CSV String of NSTR UINT16 values</td>
<td>Zero-referenced indices</td>
</tr>
</tbody>
</table>

**XML:**

```xml
<LUMP NMXSTR="4" STRIDX="0,1,2,3"></LUMP>
```
Chunk Definition

In general, the sample packing scheme must be known to correctly decode the samples. For example, consider 16 1-bit real samples packed into a UINT16 words written in little-endian format. These samples would be decoded incorrectly if shifted out from a UINT32 word (due to little-endianness).

Another consideration is whether bits are packed from left to right or vice versa.

A chunk is defined as a segment of data consisting of one or more lumps that have been packed using one of four standard unsigned integer data types.

The Stream, Lump and Chunk parameters together, completely and unambiguously describe the sample decoding scheme.
# Chunk Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Enumeration</th>
<th>Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>number of lumps</td>
<td>NLMP</td>
<td></td>
<td>UINT8</td>
<td>lumps</td>
</tr>
<tr>
<td>packed word type</td>
<td>TYPE</td>
<td>‘UINT8’, ‘UINT16’, ‘UINT32’, ‘UINT64’</td>
<td>String</td>
<td>N/A</td>
</tr>
<tr>
<td>number of words</td>
<td>NWORDS</td>
<td></td>
<td>UINT8</td>
<td>words</td>
</tr>
<tr>
<td>word endianness</td>
<td>ENDIAN</td>
<td>‘L’ – little endian ‘B’ – big endian ‘N’ – not applicable</td>
<td>string</td>
<td>N/A</td>
</tr>
<tr>
<td>Padding</td>
<td>PAD</td>
<td>‘H’ – head padding ‘T’ – tail padding ‘N’ – no padding</td>
<td>string</td>
<td>bits (value implied)</td>
</tr>
<tr>
<td>word shift direction</td>
<td>SHIFT</td>
<td>‘L’ – left shift ‘R’ – right shift</td>
<td>string</td>
<td>N/A</td>
</tr>
</tbody>
</table>

**XML:**

```xml
<-- Chunk Parameters -->
<CHUNK NLMP="1" TYPE="UINT32" NWORDS="1" ENDIAN="L" PAD="N" SHIFT="R"></CHUNK>
```
Chunk Examples

Examples: Chunk with single lump encoded within single UINT8 word. Lump:

```
Chunk Parameters:
NLMP: 1
TYPE: UINT8
WORDS: 1
ENDIAN: N
PAD: T
SHIFT: LEFT

Decoded Streams:

Chunk Parameters:
NLMP: 1
TYPE: UINT8
WORDS: 1
ENDIAN: N
PAD: H
SHIFT: RIGHT
```

```
Chunk Parameters:
NLMP: 1
TYPE: UINT8
WORDS: 1
ENDIAN: N
PAD: T
SHIFT: RIGHT

Decoded Streams:

Chunk Parameters:
NLMP: 1
TYPE: UINT8
WORDS: 1
ENDIAN: N
PAD: H
SHIFT: RIGHT
```
Example: Chunk with single lump encoded within single UINT16 word. Lump:

Chunk Params:
- NLMP: 1
- TYPE: UINT16
- WORDS: 1
- ENDIAN: B
- PAD: T
- SHIFT: LEFT

Decoded Streams:

Chunk

---

Chunk Params:
- NLMP: 1
- TYPE: UINT16
- WORDS: 1
- ENDIAN: L
- PAD: T
- SHIFT: LEFT

Decoded Streams:
Chunk Examples (contd.)

Example: Chunk with 10 lumps encoded within 9 UINT16 words. Lump:

Chunk Params:  
- NLMP: 10
- WORDS: 9
- PAD: N
- TYPE: UINT16
- ENDIAN: B
- SHIFT: LEFT

Decoded Streams:

Chunk: b0 b1 b0 b1 b0 b1 b0 b1 b0 b1 b0 b1 b0 b1 b0 b1

Decoded Streams:

Example: Chunk with 10 lumps encoded within 9 UINT16 words. Lump:

Chunk Params:  
- NLMP: 10
- WORDS: 9
- PAD: N
- TYPE: UINT16
- ENDIAN: L
- SHIFT: LEFT

Decoded Streams:
Frame Definition

A data file may contain other data in addition to SDR data that are multiplexed together. Or, the SDR data may be contained in a payload of a packet (example: when a VITA-59 stream is written to file). At a minimum, the standard must contain information to skip over bytes and point to the start of one or more chunks to be decoded. This is accomplished by defining a frame.

- A frame is comprised of a finite integer number of chunks greater than zero.
- Chunks within a frame are sequential and contiguous.
- A frame may begin with a data segment of arbitrary size (integer number of bytes) known as a header.
- A frame may end with a data segment of arbitrary size (integer number of bytes) known as a footer.
- A frame may contain data integrity features that are implemented within the header and/or footer segments.
- The frame data structure shall be constant for the data collection session (i.e. frame format shall not change dynamically).
Valid Frames

```
  ... R_0  R_1  R_2  ... R_N ... 
    chunk

  ... Header R_0  R_1  R_2  ... R_N ... 

  ... R_0  R_1  R_2  ... R_N  Footer ... 

  ... Header R_0  R_1  R_2  ... R_N  Footer ... 
```
# Frame Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Enumeration</th>
<th>Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>total number of bytes in frame</td>
<td>NUMBYTES</td>
<td></td>
<td>UINT32</td>
<td>bytes</td>
</tr>
<tr>
<td>number of bytes in header</td>
<td>HEADERBYTES</td>
<td></td>
<td>UINT32</td>
<td>bytes</td>
</tr>
<tr>
<td>number of bytes in footer</td>
<td>FOOTERBYTES</td>
<td></td>
<td>UINT32</td>
<td>bytes</td>
</tr>
<tr>
<td>number of chunks in frame</td>
<td>NUMCHUNKS</td>
<td></td>
<td>UINT32</td>
<td></td>
</tr>
</tbody>
</table>

**XML:**

```xml
<!-- Frame Parameters -->
<FRAME NUMBYTES="7044" HEADERBYTES="0" FOOTERBYTES="4" NUMCHUNKS="56320"></FRAME>
```
SDR data collection systems write formatted data into files in numerous ways. We refer to these as system topologies. The main topologies are:

- Single ‘stream of data’ written to a single file
- Single ‘stream of data’ containing two or more multiplexed streams written to a single file
- Two or more ‘streams of data’ written to individual files (each containing one stream, referred to as ‘spatial splitting’)
- Two or more ‘streams of data,’ each containing two or more multiplexed streams written to individual files
- The term ‘file’ above may be one large file or multiple files split across time intervals (temporal splitting)

The ‘stream of data’ referenced above is typically written to a high-speed disk array. Sector-aligned write modes may be used for performance reasons. For this and possibly other reasons, the start of a frame may not be aligned to the beginning of a file.
Concept of a Lane (contd.) and Definition

For spatially split files, each metadata file must identify its paired SDR file’s association with respect to other SDR files in the data set.

Spatially split files may be written across multiple computer systems. Hence, the time offsets between these systems may be needed to synchronize streams.

These time offsets may not be known at collection time but may be resolved later. Hence the format must support back-annotation.

The ‘stream of data’ terminology used above is henceforth termed ‘Lane’

A Lane is defined as a conduit that transports formatted data.
### Lane Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Enumeration</th>
<th>Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>total number of lanes in data collection system</td>
<td>NUMLANES</td>
<td></td>
<td>UINT16</td>
<td>Lanes</td>
</tr>
<tr>
<td>Lane index associated with this metadata file</td>
<td>LANEINDEX</td>
<td></td>
<td>UINT32</td>
<td>zero-referenced index</td>
</tr>
<tr>
<td>Time offset of this lane w.r.t. Lane Index 0</td>
<td>TOFFSETMS</td>
<td></td>
<td>INT32.UINT32 (N/D)</td>
<td>milliseconds</td>
</tr>
</tbody>
</table>

**XML:**

```xml
<!-- Lane Parameters -->
<NUMLANES>1</NUMLANES>
<LANEINDEX>0</LANEINDEX>
<TOFFSETMS>0.1</TOFFSETMS>
```
File Definition and Parameters

A file is defined as the ordered collection of bytes retrieved from a single Lane over a finite interval of time that is stored in a digital media device.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Type</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer byte offset to start of first frame of Lane</td>
<td>BYTEOFFSET</td>
<td>UINT32</td>
<td>Bytes</td>
</tr>
<tr>
<td>Time that this file was created</td>
<td>CREATIONTIME</td>
<td>TimeCSV string</td>
<td></td>
</tr>
<tr>
<td>Previous file name of temporally split sequence</td>
<td>PREVIOUSFILE</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>Next file name</td>
<td>NEXTFILE</td>
<td>String</td>
<td></td>
</tr>
</tbody>
</table>

XML:

```xml
<BYTEOFFSET>1234</BYTEOFFSET>
<CREATETIME>2014,12,31,23,59,59,999</CREATETIME>
<preVIOUSFILE></PREVIOUSFILE>
<NEXTFILE>WideBand_1bit_L1L2R1R2_001.dat</NEXTFILE>
```
Nice to Haves

- Ref. oscillator type and clock model (af0, af1, af2)
- SVs in view. Type, PRN, Doppler, CodePhase
- Orbit info for SVs in view. Embedded TLE data
- Events (such as anomalous events)
- Front-end gain settings/ change events
- Data/Metadata association check (e.g. MD5 hash)
Documentation Subcommittee

• Committed to all WG documentation activities
  – Especially Sept 15 to Oct 15 2014
• 2-3 persons (in addition to co-chairs)
• Responsibilities
  – Review related standards to adopt/maintain ‘best practices’ (ION Std. 101, VITA-49, IEEE standards, etc.)
  – Create/review presentations, reports and draft standard documents
  – Ensure quality of all ION SDR WG publications
• Qualifications
  – Excellent writing/editing skills
  – (preferred) Previous experience/involvement writing standards
  – (preferred) Individuals with recognized track record of publication excellence
Reference Implementation Subcommittee

• Open source implementation of metadata ‘codec’
  – Goes a long way towards successful adoption
  – Languages: C/C++, Python, MATLAB (MEX)
  – Scope: Structure(s) <-> Metadata file(s)
  – Based on free OS XML libraries (e.g. TinyXML)
  – Public repository (on GitHub, SourceForge)

• Coding group: 2-3 persons (balanced representation)

• Responsibilities:
  – Develop metadata ‘codec’ implementation according to draft standard and software development best practices
  – Has authority to revise XML format for implementability as needed
  – Support industry-standard ports and tool-flows (gcc, Make, MSVS, etc.)
  – Create and maintain public repository (gatekeeper)
How you can get involved

• Join the working group / online discussion group
• Join Documentation Subcommittee
• Join Reference Implementation Subcommittee
• Send us your SDR files and format descriptions (upload)
• Sign up for alpha testing
• **Oct 13**: Initial Draft Standard on ION.org
• **Oct 13, 2014 to Jan 12, 2015**: 90-day Open Public Comment Period
• **ION ITM 2015**: Next Working Group Meeting

Contact WG Co-Chairs:
Dr. Sanjeev Gunawardena: sanjeev@ChameleonChips.com
Dr. Thomas Pany: t.pany@ifen.com