SNEAK (inStrumeNt sourcE pAcket toolkit)

Description and Usage Examples

17-05-2022

SNEAK

- SNEAK is designed to be a toolkit to extract/manipulate the values of specific ISP fields in L0/RAW files
  - Multiple *command line utilities*
    - Could easily be used for batch processing of L0 files

  - *ISP Tree Maker*
    - lists the fields of the ISP(s) stored in a file
  - *ISP Extractor*
    - extracts specific ISP field(s) into a CSV text file
  - *ISP Transform*
    - updates specific ISP field(s) according to a timeline
  - *ISP Multiplexer*
    - selects/reorders/combines the ISPs from multi-file ISP streams
  - *ISP Sequencer*
    - corrects the SSC of a multi-file ISP stream
  - *ISP CRC Fix*
    - corrects the CRC of a multi-file ISP stream

SNEAK

- Available for Linux, MacOS and Windows (64bit)
- Documentation included in the installation package – also available online.
- Usage Requirements
  - 25 MB disk space
  - 8 GB RAM
  - Java 17, download from https://adoptium.net/
- Installation
  - Make sure JAVA_HOME environment variable is correctly set
  - Download from https://eop-cfi.esa.int/index.php/applications/tools/sneak
  - Extract anywhere
  - (Optional) Add the installation folder to the PATH environment variable

SNEAK

- Specification of L0 data format based in *DFDL for Space* (DFDL4S)
  - DFDL schema is a generic extension of XSD schema (used to validate XML)
  - DFDL schema is extends XSD with attributes/annotations that specify size/characteristics of binary fields
  - DFDL4S is *"our"* implementation of DFDL
    - available at https://eop-cfi.esa.int/index.php/applications/dfdl4s
    - Supports interpretation of L0/RAW Space-to-Ground data
    - Support CCSDS format, including Space Packets Protocol for Instrument Source Packets (ISPs)
  - Catalogue of supported mission is available at

    https://eop-cfi.esa.int/Repo/PUBLIC/DOCUMENTATION/MISSION_DATA/TELEMETRY_SCHEMA_FILES/
  - Note: the amount of detail included in the DFDL4S schema is (mostly) guided by the needs of the mission

SNEAK . *Generic*

- How to display version information?

```
$ isp_treemk --version

isp_treemk (sneak v1.3.2)
Developed and distributed by EOP-PE
https://eop-cfi.esa.int/
```

- How to display available options?

```
$ isp_treemk --help

General:
  -h [ --help ]         Display options help
  -v [ --version ]      Display version information
  --verbose             Display progress bar while processing the file
  --schema arg          Schema of the ISP [REQUIRED]
  --isp arg             ISP file to be displayed [REQUIRED]
  --output arg          Output file to store the tree [REQUIRED]
  --packet arg          Number of an ISP to be displayed [OPTIONAL, MULTIPLE]
```

- Arguments via file option

```
$ isp_treemk --via arguments.txt
```

```
schema=MissionX-bandTMISP.xsd
isp=datafile.bin
packet=1
output=tree.txt
```

SNEAK . *ISP Tree Maker*

- The ISP Tree Maker (isp_treemk) lists the fields of ISP(s) stored in a L0 file.

- Useful to collect the ISP field paths used to configure the other utilities.
  - Consider using S2G to do elaborate ISP inspection/reporting
    – available at https://eop-cfi.esa.int/index.php/applications/s2g-data-viewer

- *Options*

```
-h [ --help ]          Display options help
-v [ --version ]       Display version information
--verbose              Display progress bar while processing the file
--schema arg           Schema of the ISP [REQUIRED]
--isp arg              ISP file to be displayed [REQUIRED]
--output arg           Output file to store the tree [REQUIRED]
--packet arg           Number of an ISP to be displayed [OPTIONAL, MULTIPLE]
```

SNEAK . *ISP Tree Maker*

- How to display all packets in ISP stream?

```
$ isp_treemk
    --schema schema/FLEXX-bandTMISP.xsd
    --isp L0/FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL
```

- How to display a specific packet in ISP stream?

```
$ isp_treemk
    --schema schema/FLEXX-bandTMISP.xsd
    --isp L0/FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL
    --packet 0
```

- How to display multiple specific packets in ISP stream?

```
$ isp_treemk
    --schema schema/FLEXX-bandTMISP.xsd
    --isp L0/FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL
    --packet 5
    --packet 8
```

```
ISP#: 1
Depth 0 :: /ISP
Depth 1 :: /ISP/Packet_Primary_Header
Depth 2 :: /ISP/Packet_Primary_Header/Packet_Version == '000.....' <unsignedByte>
Depth 2 :: /ISP/Packet_Primary_Header/Packet_Identification
Depth 3 :: /ISP/Packet_Primary_Header/Packet_Identification/Packet_Type == '...0....' <unsignedByte>
Depth 3 :: /ISP/Packet_Primary_Header/Packet_Identification/Secondary_Header_Flag == '....1...' <unsignedByte>
Depth 3 :: /ISP/Packet_Primary_Header/Packet_Identification/APID
Depth 4 :: /ISP/Packet_Primary_Header/Packet_Identification/APID/PID == '.....001 0110....' <unsignedByte>
Depth 4 :: /ISP/Packet_Primary_Header/Packet_Identification/APID/PCAT == '....1001' <unsignedByte>
Depth 2 :: /ISP/Packet_Primary_Header/Packet_Sequence_Ctrl
Depth 3 :: /ISP/Packet_Primary_Header/Packet_Sequence_Ctrl/Sequence_Flags == '11......' <unsignedByte>
Depth 3 :: /ISP/Packet_Primary_Header/Packet_Sequence_Ctrl/SSC == '1' <unsignedShort>
Depth 2 :: /ISP/Packet_Primary_Header/Packet_Data_Length == '189' <unsignedShort>
Depth 1 :: /ISP/Packet_Data_Field
Depth 2 :: /ISP/Packet_Data_Field/NAVATT_Packet_Secondary_Header
Depth 3 :: /ISP/Packet_Data_Field/NAVATT_Packet_Secondary_Header/PUS_Header
Depth 4 :: /ISP/Packet_Data_Field/NAVATT_Packet_Secondary_Header/PUS_Header/Spare == '0.......' <unsignedByte>
Depth 4 :: /ISP/Packet_Data_Field/NAVATT_Packet_Secondary_Header/PUS_Header/PUS_Version == '.001....' <unsignedByte>
Depth 4 :: /ISP/Packet_Data_Field/NAVATT_Packet_Secondary_Header/PUS_Header/Spare == '....0000' <unsignedByte>
Depth 4 :: /ISP/Packet_Data_Field/NAVATT_Packet_Secondary_Header/PUS_Header/Service_Type == '3' <unsignedByte>
Depth 4 :: /ISP/Packet_Data_Field/NAVATT_Packet_Secondary_Header/PUS_Header/Service_Subtype == '25' <unsignedByte>
Depth 3 :: /ISP/Packet_Data_Field/NAVATT_Packet_Secondary_Header/Destination_Id == '00000000' <unsignedByte>
```

SNEAK . *ISP Extractor*

- The ISP Extractor (isp_extractor) extracts the values of ISP field(s) into a CSV text file
- List of fields to extract is provided in text file (one field per line):

```
/Packet_Primary_Header/Packet_Identification/APID int
/Packet_Primary_Header/Packet_Sequence_Ctrl/SSC
/Packet_Data_Field/NAVATT_Packet_Secondary_Header/Time_Code_Field/Time_Code
/Packet_Data_Field/NAVATT_Packet_Secondary_Header/Time_Code_Field/Time_Code/Coarse_Time
/Packet_Data_Field/NAVATT_Packet_Secondary_Header/Time_Code_Field/Time_Code/Fine_Time
/Packet_Data_Field/NAVATT_User_Data_Field/ISP_Data/Navigation_and_Attitude_Data/Satellite_Position/X
/Packet_Data_Field/NAVATT_User_Data_Field/ISP_Data/Navigation_and_Attitude_Data/Satellite_Position/Y
/Packet_Data_Field/NAVATT_User_Data_Field/ISP_Data/Navigation_and_Attitude_Data/Satellite_Position/Z
/Packet_Data_Field/NAVATT_User_Data_Field/ISP_Data/Navigation_and_Attitude_Data/Satellite_Velocity/X
/Packet_Data_Field/NAVATT_User_Data_Field/ISP_Data/Navigation_and_Attitude_Data/Satellite_Velocity/Y
/Packet_Data_Field/NAVATT_User_Data_Field/ISP_Data/Navigation_and_Attitude_Data/Satellite_Velocity/Z
```

- *Options*

  -h [-help] : Displays options help
  -v [-version] : Displays version information
  --verbose : Display progress bar while processing the file
  --schema arg : Schema of the ISP [REQUIRED]
  --isp arg : ISP file [REQUIRED]
  --fields arg : Text file containing the list of fields (one per line) to be extracted [REQUIRED]
  -o arg ` [ `-output arg] : Output file to store the CSV fields [REQUIRED]
  --separator arg : The separator to be used in the output; if not provided ‘,’ is used by default [OPTIONAL]
  --unavailable arg : The [integer] value to be used when a field does not exist; if not provided ‘0’ is used by default [OPTIONAL]

SNEAK . *ISP Extractor*

- How to extract ISP field values into CSV?

```
$ isp_extractor
    --schema schema/FLEXX-bandTMISP.xsd
    --isp L0/FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL
    --fields cfg/flex_fields.txt
    --output flex_fields.csv
   [--verbose]
```

- How to extract ISP field values with custom separator?

```
$ isp_extractor
    --schema schema/FLEXX-bandTMISP.xsd
    --isp L0/FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL
    --fields cfg/flex_fields.txt
    --output flex_fields.csv
    --separator “;”
```

- How to extract ISP field values with custom “unavailable field marker”?
  - *This is typically useful in case of processing ISP streams mixing multiple ISP types, where some types might not have all selected fields*

```
$ isp_extractor
    --schema schema/FLEXX-bandTMISP.xsd
    --isp L0/FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL
    --fields cfg/flex_fields.txt
    --output flex_fields.csv
    --separator “;”
    --unavailable 42
```

```
#,/Packet_Primary_Header/Packet_Identification/APID,/Packet_Primary_Header/Packet_Sequence_Ctrl/SSC,/Pa
1,01 69,16380,2019-09-14T10:36:19.470
2,01 69,16381,2019-09-14T10:36:17.470
3,01 69,16382,2019-09-14T10:36:15.470
4,01 69,16383,2019-09-14T10:36:13.470
5,01 69,0,2019-09-14T10:36:14.470
6,01 69,1,2019-09-14T10:36:16.470
7,01 69,2,2019-09-14T10:36:18.470
8,01 69,3,2019-09-14T10:36:20.470
```

SNEAK . *ISP Transform*

- The ISP Transform (isp_transform) updates specific ISP field(s) according to a timeline specification
- Traverses all ISPs in the provided ISP stream, and updates the fields based on the ISP packet time information

Warning: The changes are applied to the ISP stream ”in-situ”!!! The ISP file will be changed!

- *Options:*

```
-h [ --help ] : Displays options help
-v [ --version ] : Displays version information
--verbose : Display progress bar while processing the file
--schema arg : Schema of the ISP [REQUIRED]
--isp arg : ISP file [REQUIRED]
--timeline arg : Timeline file that defines the ISP fields to be updated [REQUIRED]
```

SNEAK . *ISP Transform*

Degraded L0 Use Case: Packet Corruption

- Define a “timeline” for corruption

```
{
  "time_segments": [
    {
      "time_selection": {
        "path": "/Packet_Data_Field/NAVATT_Packet_Secondary_Header/Time_Code_Field/Time_Code/Coarse_Time",
        "gps_time_start": "2019-09-14T10:36:33.000",
        "gps_time_stop": "2019-09-14T10:36:34.000",
      },
      "updates": [
        {
          "path": [
            "/Packet_Data_Field/NAVATT_User_Data_Field/Packet_Error_Control/CRC”
          ],
          "type": "uint16_t",
          "value": ”42"
        }
      ]
    }
  ]
}
```

- Introduce the corruption in the L0 Packet stream with isp_transform

```
$ isp_transform
    --schema schema/FLEXX-bandTMISP.xsd
    --isp L0/FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL
    --timeline cfg/flex_timeline.json
   [--verbose]
```

SNEAK . *ISP Multiplexer*

- The ISP Multiplexed (isp_mux) select/reorder/combine the ISPs from multi-file ISP streams

```
{
  "order": [
    {
      "schema_path": "schema/FLEXX-bandTMISP.xsd",
      "isp_path": "FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL",
      "isp_indexes": [
        2,
        0
      ]
    },
    {
      "schema_path": "schema/FLEXX-bandTMISP.xsd",
      "isp_path": "FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL",
      "isp_indexes": [
        1,
        3
      ]
    }
  ]
}
```

- Options:

```
 -h [ --help ]          Display options help
 -v [ --version ]       Display version information
 --verbose              Display progress bar while processing the file
 -o [ --output ] arg    Output file to store the selected ISPs [REQUIRED]
 --order arg            Path to configuration describing order of ISP [REQUIRED]
```

SNEAK . *ISP Multiplexer*

Degraded L0 Use Case: Gap across single/multiple packets

- Create a configuration of the intended order

```
{
  "order": [
    {
      "schema_path": "schema/FLEXX-bandTMISP.xsd",
      "isp_path": "FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL",
      "isp_indexes": [
        0, 1, 2, 3, , 5, 6, 7, 8, 9, 10, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 31, 33, 34, 35, …
      ]
    } ]
}
```

Degraded L0 Use Case: Stuttering/Repeated/Overwritten packets

- Create a configuration of the intended order

```
{
  "order": [
    {
      "schema_path": "schema/FLEXX-bandTMISP.xsd",
      "isp_path": "FLX_GPP__L0__NAVATT_20190914T103613_20190914T103728_20220201T153727.DBL",
      "isp_indexes": [
        0, 1, 2, 3, 4, 4, 4, 4, 5, 6, 5, 7, 5, 8, 5, 9, 10, 11, 12, 13, 14, 15, 10, 11, 12, 19, 20, …
      ]
    }
  ]
}
```

- Create new ISP stream with intended packet sequence using isp_mux

```
$ isp_mux
    --order cfg/flex_order.json
    --output flex_isp.bin
```

SNEAK . *ISP Sequencer*

- The ISP Sequencer (isp_sequencer) corrects the SSC of a multi-file ISP stream

- Traverses all ISPs from the multiple provided ISP streams, making the SSC sequential
  - SSC sequence is independent per APID
    - i.e. it can be used on a ISP stream containing multiple types of packets
  - Only the SSC is updated, so the CRC needs to be adjusted as a second step (see isp_crcfix tool)

- Options:

```
 -h [ --help ]         Display options help
  -v [ --version ]      Display version information
  --verbose             Display progress bar while processing the file
  --schema arg          Schema of the ISP [REQUIRED]
  --isp arg             ISP(s) file to be displayed [REQUIRED]
  --ssc arg             Pair(s) of <APID>,<APID>:<SCC> (e.g 1343,1353:42); if not provided initial SSC defaults to 0
```

SNEAK . *ISP Sequencer*

- How to make a correct SSC sequence in ISP stream?

```
$ isp_sequencer
    --schema schema/FLEXX-bandTMISP.xsd
    --isp flex_isp.bin
   [--verbose]
```

- How to make a correct SSC sequence in a multiple file ISP stream?

```
$ isp_sequencer
    --schema schema/FLEXX-bandTMISP.xsd
    --isp flex_isp_1.bin
    --isp flex_isp_2.bin
```

- How to make a correct SSC sequence in ISP stream with custom starting SSC?

```
$ isp_sequencer
    --schema schema/FLEXX-bandTMISP.xsd
    --isp flex_isp_1.bin
    --isp flex_isp_2.bin
    --ssc 1343,1353:16382
```

SNEAK . *ISP CRC Fix*

- The ISP CRC Fix (isp_crcfix) corrects the CRC of all packets in multi-file ISP stream

- Traverses all ISPs from the multiple provided ISP files, correcting the CRC to ensure that ISP is fully valid
  - It is assumed that: the CRC is standard and is the last field described by the schema

- Options:

```
 -h [ --help ]         Display options help
  -v [ --version ]      Display version information
  --verbose             Display progress bar while processing the file
  --schema arg          Schema of the ISP [REQUIRED]
  --isp arg             ISP(s) file to be displayed [REQUIRED]
```

SNEAK . *ISP CRC Fix*

- How to correct all CRCs in one ISP stream?

```
$ isp_crcfix
    --schema schema/FLEXX-bandTMISP.xsd
    --isp flex_isp.bin
   [--verbose]
```

- How to correct all CRCs in a multiple file ISP stream?

```
$ isp_crcfix
    --schema schema/FLEXX-bandTMISP.xsd
    --isp flex_isp_1.bin
    --isp flex_isp_2.bin
```

Support

- In-house development
  - Quick feedback
  - Quick releases
- Active Maintenance/Support

- Suggestions of new features, improvements or bug reports are welcome!

- Support:
  cfi@eopp.esa.int or dfdl4s@eopp.esa.int