Installation Instructions for the Community Satellite Processing Package (CSPP) VIIRS, ATMS, and CrIS SDR Version 3.1.1 Software for Suomi NPP and NOAA-20 (JPSS-1)

University of Wisconsin-Madison, Space Science and Engineering Center (SSEC)
Supported by the NASA/NOAA Joint Polar Satellite System (JPSS) Project

17 October 2018
5 November 2018 - New CrIS and ATMS BUFR conversion scripts

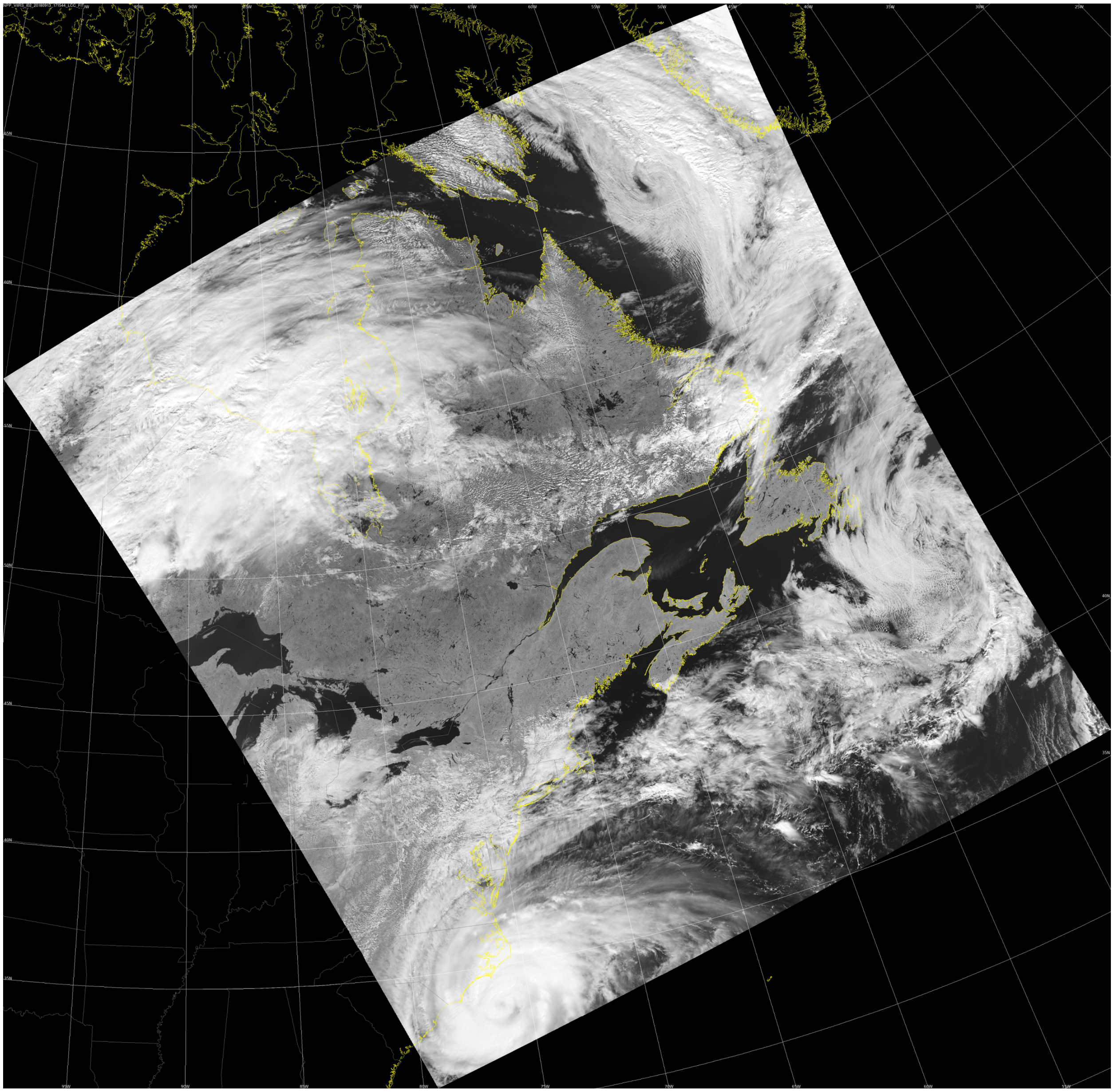


Table of Contents

Section 1: Introduction

1.1 Overview

This document contains instructions for installation and operation of the Community Satellite Processing Package (CSPP) software for transforming direct broadcast Suomi NPP and NOAA-20 VIIRS, ATMS, and CrIS Raw Data Record (RDR) files to Sensor Data Record (SDR) products on Intel Linux computers. The CSPP SDR software package contains binary executable files and supporting static data files, as well as input and output files for verification of a successful installation.

The CSPP SDR software package is based on the Algorithm Development Library (ADL) software developed by Raytheon for the JPSS Project. ADL allows the operational processing algorithms for Suomi NPP and NOAA-20 (JPSS-1) to run without modification in a Linux environment. SSEC has packaged the ADL versions of the algorithms so they can run from the Linux command line in real-time direct broadcast mode, but we have not changed the underlying processing software, algorithms, or data formats. Source code is not included with this package; it is available separately from SSEC as part of the ADL distribution. The output files created by this software are identical in naming, format (HDF5), and structure to the corresponding files from the NOAA/NESDIS CLASS archive. Descriptions of the SDR file formats are available in the Common Data Format Control Book (CDFCB) External Vol. III at:

https://jointmission.gsfc.nasa.gov/documents.html

The CSPP SDR software is distributed through the CSPP website at:

http://cimss.ssec.wisc.edu/cspp/

Software, test data, and documentation may be downloaded from this web site. Please use the 'Contact Us' form on the website to submit any questions or comments about CSPP.

1.2 What's New in CSPP SDR Version 3.1 and Version 3.1.1

- **v3.1.1** - Fixed bugs in v3.1 which resulted in reduced granule yield.
- CSPP SDR v3.1 code base updated to Block 2.1 Mx2 (ADL5.3_I2.1.02).
- Filtering is now performed by default to remove incomplete or partial RDRs prior to processing. This behavior can be controlled using option -s.
- Refined Look-Up-Table (LUT) and ancillary data download and storage.
- Fixed bug that repeated storage of LUT and ancillary data sets into the database.
- New ancillary and LUT distribution site is used: http://jpssdb.ssec.wisc.edu/cspp_v_3_1/

1.3 System Requirements

System requirements for the CSPP SDR software are as follows:

- Intel or AMD CPU with 64-bit instruction support,
- 16 GB RAM (minimum), plus an additional 8GB of RAM for every core that is used for CSPP VIIRS SDR processing.

Please note: S-NPP and the final NOAA-20 (JPSS-1) satellite orbits are 50 minutes apart. In order to keep up with processing data from both satellites, **we strongly recommend** multi-core processing (sdr -p option), and increasing your system memory if necessary.

- CentOS 6.9 64-bit Linux (or other compatible 64-bit Linux distribution),
- 100 GB disk space (minimum),
- Internet connection (for downloading ancillary data).

Linux terminal commands included in these instructions assume the bash shell is used.

1.4 Input Data Requirements

VIIRS, CrIS, and ATMS Raw Data Record (RDR) files in HDF5 format are required as input to the CSPP SDR v3.1 processing software. RT-STPS version 5.9 or higher from the NASA Direct Readout Laboratory creates RDR files that can be ingested by the CSPP SDR processing software.

1.5 Disclaimer

Original scripts and automation included as part of this package are distributed under the GNU GENERAL PUBLIC LICENSE agreement version 3. Binary executable files included as part of this software package are copyrighted and licensed by their respective organizations, and distributed consistent with their licensing terms.

The University of Wisconsin-Madison Space Science and Engineering Center (SSEC) makes no warranty of any kind with regard to the CSPP software or any accompanying documentation, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. SSEC does not indemnify any infringement of copyright, patent, or trademark through the use or modification of this software.

There is no expressed or implied warranty made to anyone as to the suitability of this software for any purpose. All risk of use is assumed by the user. Users agree not to hold SSEC, the University of Wisconsin-Madison, or any of its employees or assigns liable for any consequences resulting from the use of the CSPP software.

Section 2: Implementation and Configuration

2.1 Overview

This software package contains the JPSS Algorithm Development Library (ADL) version 5.3_I2.1.02 codebase VIIRS, ATMS, and CrIS SDR algorithms bundled together as a standalone processing system. The SDR algorithms correspond to the Block 2.1 Mx2 version running in the Interface Data Processing Segment (IDPS), NOAA's current operational system. The software is compiled for Intel Linux 64-bit ("x86_64") systems compatible with CentOS 6.9. The software reads RDRs and writes SDRs in a HDF5 format compatible with the JPSS Common Data Format Control Books (CDFCBs).

2.2 CSPP SDR Version 3.1 Implementation

- This release replaces existing installations of CSPP SDR (it is not an update). The CSPP SDR v3.1 executable, and static tarfiles **must be used** for this release. The tarfiles from previous releases cannot be reused.
- This release is based on ADL v5.3_I2.1.02, and includes software changes derived as a result of Suomi-NPP and NOAA-20 Calibration/Validation efforts.
- A set of release notes on general and specific instrument updates and changes is provided in the SDR_3_1/RELEASE_NOTES.txt document.

2.2.1 Processing Historical Data Sets with CSPP SDR Version 3.1

CSPP SDR v3.1 can be used for creating VIIRS, CrIS and ATMS SDRs from historical S-NPP RDRs beginning with data collected on 8 August 2017, and NOAA-20 RDRs beginning on 1 February 2018.

2.2.2 Direct Broadcast Data Truncation

To accommodate the transmission of CrIS Full Spectral Resolution (FSR) observations, S-NPP VIIRS M-Band 7 is not included in the High Rate Data (HRD) (direct broadcast) transmission. In addition, the S-NPP satellite does not include CrIS Field-of-View (FOV) number 4 (beginning 4 December 2014) and number 6 (beginning 8 August 2015) in the HRD. The un-truncated data sets are available from the NOAA CLASS archive.

As in the previous CSPP SDR versions, the CSPP V3.1 software handles this data deletion with the following approach:

CrIS SDR
S-NPP CrIS Full Spectral Resolution **(**FSR) and Normal Spectral Resolution (NSR) SDRs contain fill values for all missing FOVs.

VIIRS SDR
S-NPP and NOAA-20 downlink of CrIS FSR data comes at the expense of the downlink of M-Band 7 (0.865 micron). CSPP SDR v3.1 reconstructs VIIRS M7 using data from VIIRS I2 if the SDR software detects that the M7 data set contains fill values, while the I2 data contains valid data. The reconstructed M7 data is very similar (but not identical) to the M7 data that was not transmitted. The algorithm used by CSPP SDR v3.1 for duplication is very simple. It computes the average of the overlapping I2 FOVs to create the M7 replacement FOV. If any of the I2 data values that go into producing one M7 FOV is missing, a missing value is stored in the M7 output FOV. Since M7 is a dual gain channel, differences in the reconstructed array may be most noticeable in dark (small radiance/reflectance) scenes. Downstream products which use of M7 reflectances will be affected by this as well, including the VIIRS cloud mask product.

In addition, this reconstruction requires the setting of quality flags for the newly created M7 SDR. Since the format of the quality flags for both I-bands and M-Bands are identical, combining the 4 SVI02 data quality values into the final SVM07 is done using the logical "OR" function. If any of the bits that go into a quality flag value is set (1) for any of the SVI02 pixels, then the reconstructed SVM07 quality flag bit will also be set. Although this may not perfectly preserve all of the quality information stored in the SVI02 quality flag arrays for all cases, it will be enough to alert users of potential problems with a given Field-of-View.

ATMS SDR
The CSPP ATMS software is not affected by the downlink truncation.

2.2.3 NOAA-20 Direct Broadcast Data Dropouts and CSPP Filtering

Transmission of the full suite of VIIRS, CrIS and ATMS on NOAA-20 periodically results in data dropouts primarily over land during the day. These dropouts result in packet loss at local receiving stations, and at times adversely affects the calibration of ATMS and CrIS. Therefore, the CSPP SDR software now filters partial spacecraft diary and science RDRs for ATMS and CrIS instruments based upon statistics of standard APID sizes. The end effect of the added filtering is reduced output granule yields; however this guarantees the direct broadcast CSPP SDR radiances match the operational NOAA IDPS CrIS and ATMS radiances.

The CSPP software allows the user to turn off the filtering by using the -s option to cris_sdr.sh and atms_sdr.sh, however we do not guarantee compliance with IDPS calibrated products if this option is used.

Section 3: Installation of VIIRS, ATMS and CrIS SDR

3.1.1 New CSPP SDR Version 3.1.1 Installation

Download the following files from the CSPP website site:

http://cimss.ssec.wisc.edu/cspp/

```
CSPP_SDR_V3.1.tar.gz
CSPP_SDR_V3.1_static_luts.tar.gz
CSPP_SDR_V3.1_static_tiles.tar.gz
CSPP_SDR_V3.1.1_patch.tar.gz
```

Install the software as shown below (a new directory named `SDR_3_1` will be created). In this example, the tar files are assumed to be in the user’s home directory.

```
mkdir CSPP
cd CSPP
tar xf ../CSPP_SDR_V3.1.tar.gz
tar xf ../CSPP_SDR_V3.1_static_luts.tar.gz
tar xf ../CSPP_SDR_V3.1_static_tiles.tar.gz (this may take a few
minutes)
```

Apply the CSPP Version 3.1.1 patch on top of your current installation.

```
tar xf ../CSPP_SDR_V3.1.1_patch.tar.gz
```

Set the CSPP_SDR_HOME environment variable to the name of the directory where CSPP SDR was installed ($HOME in this example), and then execute the environment script as shown below:

```
export CSPP_SDR_HOME=$HOME/CSPP/SDR_3_1
source $CSPP_SDR_HOME/cspp_sdr_env.sh
```

If you want to run the test case, download the following file:

```
CSPP_SDR_V3.1_TEST_DATA.tar.gz
```

The test data should be unpacked in a directory separate from the CSPP SDR installation, e.g.,

```
cd $HOME
tar xf CSPP_SDR_V3.1_TEST_DATA.tar.gz
```

3.1.2 Patching an Existing CSPP SDR Version 3.1

Download the following files from the CSPP website site:

http://cimss.ssec.wisc.edu/cspp/

```
CSPP_SDR_V3.1.1_patch.tar.gz
```

Set the CSPP_SDR_HOME environment variable to the name of the directory where CSPP SDR was installed ($HOME in this example), and then execute the environment script as shown below:

```
export CSPP_SDR_HOME=$HOME/CSPP/SDR_3_1
source $CSPP_SDR_HOME/cspp_sdr_env.sh
```

Now apply the CSPP Version 3.1.1 patch on top of your current installation.

```
cd $CSPP_SDR_HOME/..
tar xf ../CSPP_SDR_V3.1.1_patch.tar.gz
```

Confirm a successfully patched installation by checking for Version number 3.1.1 in the SDR_3_1/RELEASE_NOTES.txt file.

3.2 Updating Look-Up-Tables (LUTs)

Upon completion of CSPP SDR Version 3.1 installation, the VIIRS, ATMS, and CrIS Look-Up-Tables (LUTs) must be integrated into your local system database. This is done through the execution of the sdr_luts.sh script.

We recommend executing the script in a work directory, as it will create and remove intermediate files (include a /log directory) upon execution and completion. To download and install the latest LUT files into the local database, run the update script as shown below (an Internet connection is required):

```
sdr_luts.sh
```

Use -W to designate a work directory and -d to prevent clean-up, for example:

```
sdr_luts.sh -d -W <path to work directory>
```

Please note: The SDR processing scripts will fail until the sdr_luts.sh script is executed at least once, to initialize and populate the database with the LUT fields.

When new LUT files are posted to the CSPP LUT/ancillary distribution website, they will be downloaded and installed by `sdr_luts.sh`. Since these files are updated periodically, it is recommended that you run this script once a week to ensure that SDR products are accurately calibrated and navigated. You may want to set up a scheduled execution via cron to routinely check for updated LUT files.

3.3 Updating Dynamic Ancillary Data

The CSPP SDR software requires dynamic ancillary data matching the date and time of the SDR file within a specified tolerance. It is therefore necessary to periodically update these data files. The CSPP SDR execution scripts will automatically determine the files that are required based upon the RDR date and time. It will then check the local cache directory to see if there are any matching Two Line Element (TLE), Polar Wander (PW) and RSBAUTOCAL-HISTORY files. If it does not find matching files, it will automatically attempt to download the files from the CSPP ancillary data archive.

Alternatively, the files can be updated by using the dynamic ancillary data update script. **We recommend executing the script in a work directory, as it will create and remove intermediate files (include /log directory) upon execution and completion.** To download and install the latest ancillary data files, run the update script as shown below (an Internet connection is required):

```
sdr_ancillary.sh
```

Use -W to designate a work directory and -d to prevent clean-up, for example:

```
sdr_ancillary.sh -d -W <path to work directory>
```

We recommend using this script as the optimal method for updating ancillary data each day. Routine daily SDR ancillary script execution (for example using linux cron scheduler) will result in ensuring the best SDR processing latency. If you do not wish to query the CSPP SDR ancillary external archive for files at runtime as part of the CSPP SDR processing, use the '-l' option in the SDR processing scripts, (e.g., atms_sdr.sh -l).

By default, `sdr_ancillary.sh` will only download the most recent ancillary files. For historical data processing, use the command `sdr_ancillary.sh –t <YYYYMMDD>` if you want to populate the ancillary database for a specific date. For example,

```
sdr_ancillary.sh -t 20180826
```

will download and populate the ancillary database for a VIIRS, CrIS or ATMS dataset acquired on 26 August 2018.

Section 4: VIIRS SDR Software

4.1 VIIRS SDR Driver Script

The principal script is `SDR_3_1/viirs/viirs_sdr.sh`, which operates by default in the current directory. It creates intermediate files and soft links, as well as final HDF5 (*.h5) output files. It is recommended that all files be removed from the work directory before each execution of `viirs_sdr.sh`. To execute the VIIRS SDR processing script, you must provide the path and name of the VIIRS RDR file or files to be processed, e.g.

```
viirs_sdr.sh RNSCA-RVIRS_npp_d20180325_t1951242_e2002421_b00001.h5
```

or

```
viirs_sdr.sh /data/noaa20/RNSCA-RVIRS_j01*.h5
```

Several options are available for `viirs_sdr.sh` as shown below:

```
-p <num_procs>  Use <num_procs> number of processor cores,
-a      Aggregate output files into one file per product type (default is no aggregation),
-z      Compress HDF5 output files with transparent internal gzip (default is no compression),
-d      Retain all intermediate files created during processing (default is to delete these files),
-g      Retain ellipsoid geolocation products (GMODO and GIMGO files),
-l      Do not download ancillary files upon execution (default is to download files if needed),
-m      Increase geolocation accuracy.  Adds additional computations and latency.
–h      Print helpful information. Use this to display all available options.
```

For example:

```
viirs_sdr.sh -p 4 /data/RNSCA-RVIRS_*.h5
```

will process 4 VIIRS SDR granules at a time from all the RDRs that are found in the /data directory by using 4 processor cores simultaneously. The output files will be identical no matter how many processor cores are used.

```
viirs_sdr.sh -g /data/npp/RNSCA-RVIRS_*.h5
```

will create VIIRS SDR h5 files from all RDR files found in the `/data/npp` directory, and include the creation of both Terrain Corrected (GITCO and GMTCO) and Ellipsoid (GIMGO and GMODO) geolocation files.

```
viirs_sdr.sh -l -a -z -p 2 RNSCA-RVIRS_npp_d20160228_t1715045_e1726273_*.h5
```

will aggregate and compress all HDF5 output files, and run using only local dynamic ancillary files. Note that when aggregation is selected, only one RDR input file may be specified on the command line. 2 processor cores will be used for creating SDRs and for aggregating the granules.

```
viirs_sdr.sh -m /data/noaa20/RNSCA-RVIRS_j01*.h5
```

Using this command will process all NOAA-20 RDR files found within the /data/noaa20 directory using more precise geolocation calculations. This adds additional processing time to the output SDR file creation.

4.2 Running the VIIRS SDR Test Case

To run the VIIRS SDR test case, unpack the test data as shown in Section 3.0 and then execute the commands below:

```
cd viirs_test
mkdir work_j01
cd work_j01
viirs_sdr.sh -p 4 ../input/RNSCA-RVIRS_j01*.h5
```

The test case will use 4 processing cores to create 7 output NOAA-20 VIIRS granules, each of 85.7 seconds duration. The processing should be completed in 5-6 minutes. For each granule, 25 HDF5 files (*_cspp_dev.h5) are created. If the VIIRS processing script runs normally, it will return a status code equal to zero. If the VIIRS processing script encounters a fatal error, it will return a non-zero status code.

To verify your output files against the output files created at NOAA operations, execute the following commands:

```
cd ..
./viirs_sdr_compare.bash idps_output_j01/ work_j01/
```

This script compares CSPP SDR I-Band and M-Band output file Radiances with operational NOAA IDPS output files for all granules. The number of differences found for each band and granule will be printed. There should be few, if any differences.

4.3 Creating VIIRS Quicklook Images

To create quicklook images of the VIIRS SDRs, use the ql_sdr.sh script:

```
$CSPP_SDR_HOME/viirs/ql_sdr.sh path_to_sdrs/
```

This script will produce an image of each VIIRS band (except the Day/Night Band) using all of the M-band and I-band SDR files found in the `path_to_sdrs/` directory. The images will be created in the current directory, e.g.,

```
ql_sdr.sh work_j01/
```

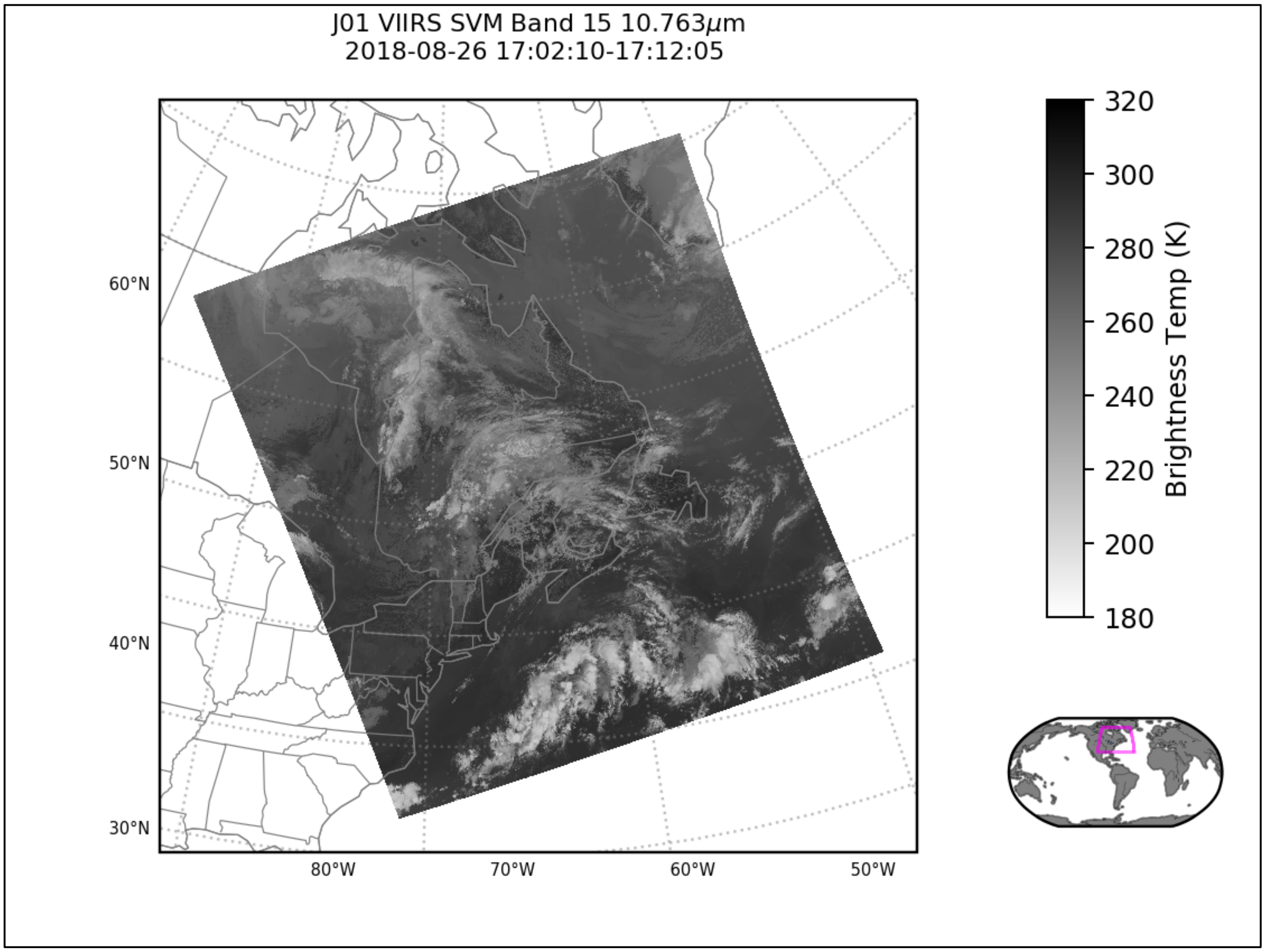
will create images of all the I-Band and M-Band files in the `work_j01/` directory and store them in the current directory. You can also create individual band images by using wildcards, e.g.,

```
ql_sdr.sh work_j01/SVM15*.h5
```

will create an aggregated PNG image of the VIIRS M-Band 15 brightness temperatures from all of the SDR granules located in the work directory, as long as the matching geolocation files are collocated. An example of the SVM15 quicklook image,

SVM15_j01_d20180826_t1703344_e1704589_b03992_c20180912183614967400_cspp_dev.png

created from the test data is shown below.



Other quick look options are available. Use ql_sdr.sh -h for a complete list.

Section 5: ATMS SDR Software

5.1 ATMS SDR Driver Script

The principal script is `SDR_3_1/atms/atms_sdr.sh`, which operates by default in the current directory. It creates intermediate files and soft links, as well as final HDF5 (*.h5) output files. It is recommended that all files be removed from the work directory before each execution of `atms_sdr.sh`. There is a cross-granule requirement for calibration of ATMS; 3 contiguous good quality science RDR granules must be available to create one output SDR granule.

To execute the ATMS SDR processing script, you must provide the path and name of the ATMS RDR file or files to be processed, e.g.

```
atms_sdr.sh RATMS-RNSCA_j01_d20180802_t1641040_e1646560_b00001.h5
```

or

```
atms_sdr.sh /data/npp/RATMS-RNSCA*.h5
```

Several options are available for `atms_sdr.sh` as shown below:

```
-p <num_procs>  Use <num_procs> number of processor cores,
-a      Aggregate output files into one file per product type (default is no aggregation),
-z      Compress HDF5 output files with transparent internal gzip (default is no compression),
-t  N N  Trim N beginning number of granules, and trim N ending number of granules in
           a sequence.
-d      Retain all intermediate files created during processing (default is to delete these files),
-l      Do not download ancillary files upon execution (default is to download files if needed),
-m      Increase geolocation accuracy.  Adds additional computations and latency.
-s      Do no apply science RDR filtering.  Caution should be exercised when using this
        option as it may result in calibration that does not match IDPS output.
-h      Print helpful information. Use this to display all available options.
```

For example:

```
atms_sdr.sh -p 2 -a -z RATMS-RNSCA_j01_d20180820_t 1708533_e1709249*.h5
```

will use 2 processor cores to generate the SDRs from the input NOAA-20 RDR, and aggregate and internally compress all HDF5 output files. Note that when aggregation is selected, only one RDR input file may be specified on the command line.

```
atms_sdr.sh -p 4 -t 2 1 /RDR/RATMS-RNSCA_npp_d20180426_t2259103_*.h5
```

Executing this command will create S-NPP ATMS SDR files from all granules contained in the RDR except the first two, and the last one in the contiguous sequence, using 4 processor cores.

5.2 Running the ATMS SDR Test Case

To run the ATMS SDR test case, unpack the test data as shown in Section 3.0 and then execute the commands below:

```
cd atms_test
mkdir work_j01
cd work_j01
atms_sdr.sh -p 4 ../input/RATMS-RNSCA_j01*.h5
```

The test case should run in less than 1 minute and create 13 output granules. The input RDR arrays are scanned for size; in this test NOAA-20 test case, 2 spacecraft diary and 2 ATMS RDR science arrays are rejected because they do not meet size thresholds.

For each granule, 3 HDF5 files will be produced (GATMO*.h5, SATMS*.h5 and TATMS*.h5). If the ATMS processing script runs normally, it will return a status code equal to zero. If the ATMS processing script encounters a fatal error, it will return a non-zero status code.

The ATMS SDR processing operates on input granules of 12 scan-lines each. The “ADL_Unpacker” is used to de-aggregate and extract processing granules from input RDR HDF5 files. For a given contiguous sequence of N raw data record (RDR) processing granules, the user can expect up to N-2 SDR granules, also of 12 scan-lines, will be produced. The first 12 scan-line processing granule and the last 12 scan-line processing granules are used for context and are not processed by the SDR algorithm. This yield may be reduced if there are incomplete intermediate granules.

If successful, the current directory will contain output in HDF5 format. Log files will be written to the "log" directory located in the working directory.

To verify your output files against the output files created at NOAA operations, execute the following commands:

```
cd ..
./atms_sdr_compare.bash idps_output_j01/ work_j01/
```

This script compares CSPP ATMS output file Brightness Temperatures and Geolocation file Latitude and Longitudes with operational NOAA IDPS output files for all granules. The number of differences found will be printed. A successful test will result in few, if any differences.

5.3 Creating ATMS Quicklook Images

To create quicklook images of all channels of the ATMS SDRs, use the ql_sdr.sh script:

```
$CSPP_SDR_HOME/viirs/ql_sdr.sh path_to_sdrs/
```

This script will produce an image of each ATMS channel (22 bands) from the SATMS*.h5 antenna temperature and GATMO*.h5 geolocation files found in the `path_to_sdrs/` directory. The output files will be created in the current directory, e.g.

```
ql_sdr.sh work_j01/
```

will create NOAA-20 ATMS antenna temperature images from files located in the `work_j01/` directory and place them in the current directory.

Using the --plot pcolormesh option produces images very quickly, but at lower resolution:

```
ql_sdr.sh --plot pcolormesh work_j01/
```

You can also create individual band images by using –c <channel number> option, e.g.,

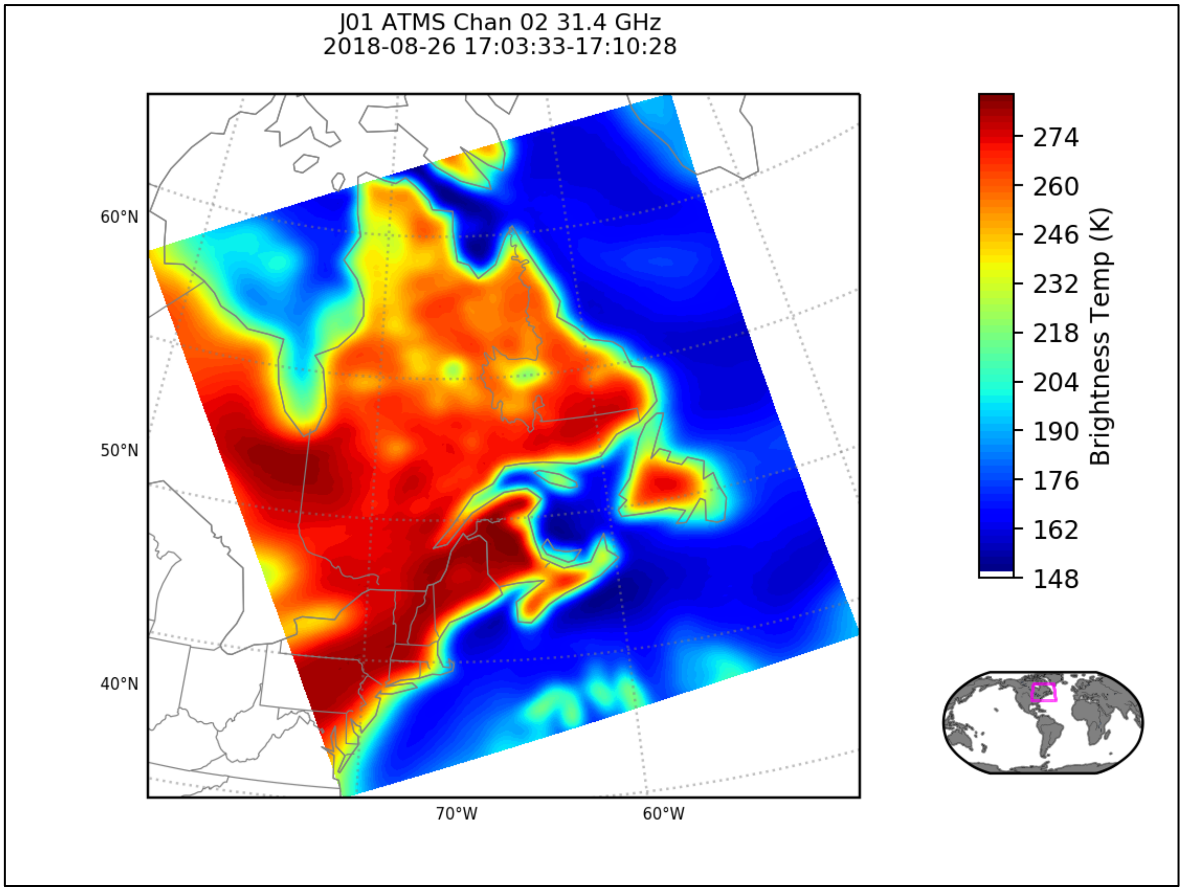
```
ql_sdr.sh -c 2 work_j01/
```

will create an aggregated PNG image of the ATMS Channel 2 antenna temperatures from all of the SDR granules located in the work directory, as long as the matching geolocation files are collocated.

An example of the NOAA-20 ATMS channel 2 quicklook image,

SATMS_2_j01_d20180826_t1703333_e1704049_b03992_c20180912204324482839_cspp_dev.png

from the test data set is shown below:



Other quick look options are available. Use ql_sdr.sh -h for a complete list.

Section 6: CrIS SDR Software

6.1 CrIS SDR Driver Script

The principal script is `SDR_3_1/cris/cris_sdr.sh`, which operates by default in the current directory. It creates intermediate files and soft links, as well as final HDF5 (*.h5) output files. It is recommended that all files be removed from the work directory before each execution of `cris_sdr.sh`. There is a cross-granule requirement for calibration of CrIS; 9 contiguous good quality science RDR granules must be available to create one output SDR granule.

Users have the option to create either the default CrIS Normal Spectral Resolution (NSR) products (SCRIS and GCRSO - 1305 apodized channels) or the CrIS Full Spectral Resolution (FSR) products (SCRIF and GCRSO - 2211 apodized channels). Use the -f flag with cris_sdr.sh to produce the FSR products, and -n to produce NSR. FSR is the default resolution for NOAA-20 CrIS and NSR is the default resolution for S-NPP CrIS.

To execute the CrIS SDR processing script, you must provide the path and name of the CrIS RDR file or files to be processed, e.g.

```
cris_sdr.sh RCRIS-RNSCA_npp_d20180416_t1641040_e1646560_b00001.h5
```

or

```
cris_sdr.sh /data/j01/RCRIS-RNSCA_j01*.h5
```

Several options are available for `cris_sdr.sh` as shown below:

`-p <num_procs>` Use <num_procs> number of processor cores,
`-f` Create the Full Spectral Resolution (FSR) product (SCRIF) (default for NOAA-20),
`-n` Create the Normal Spectral Resolution (NSR) product (SCRIS) (default for S-NPP),
`-a` Aggregate output files into one file per product type (default is no aggregation),
`-z` Compress HDF5 output files with transparent internal gzip (default is no compression),
`-d` Retain all intermediate files created during processing (default is to delete these files),
`-l` Do not download ancillary files upon execution (default is to download files if needed),
`-m` Increase geolocation accuracy. Adds additional computations and latency,
`-s` Do no apply science RDR filtering. Caution should be exercised when using this option as it may result in calibration that does not match IDPS output.
`-h` Print helpful information. Use this to display all available options.

For example:

```
cris_sdr.sh -f -p 4 /data/RCRIS-RNSCA_npp*.h5
```

will process 4 S-NPP CrIS SDR granules at a time from all the RDRs that are found in the /data directory using 4 processor cores and produce Full Spectral Resolution (FSR) output files. The output files will be identical no matter how many processor cores are used.

```
cris_sdr.sh –n -l -a -z RCRIS-RNSCA_j01_d20180228_t1716000_e1726080.h5
```

will aggregate and compress all NOAA-20 CrIS Normal Spectral Resolution (NSR) HDF5 output files, and execute using only local dynamic ancillary files. Note that when aggregation is selected, only one RDR input file may be specified on the command line.

6.2 Running the CrIS SDR Test Cases

CSPP SDR allows the creation of both CrIS Normal Spectral Resolution (NSR) and Full Spectral Resolution (FSR) output files for both S-NPP and NOAA-20 satellites.

6.2.1 Executing the CrIS FSR Test Case

To run the CrIS SDR FSR test case, unpack the test data as shown in Section 3.0 and then execute the commands below:

```
cd cris_test
mkdir work_fsr_j01
cd work_fsr_j01
cris_sdr.sh -p 4 ../input/RCRIS-RNSCA_j01*.h5
```

The test case executed using 4 processing cores should run in 6-8 minutes and create 7 output granules. The input RDR array sizes are scanned; in this test NOAA-20 test case, 2 spacecraft diary and 2 CrIS RDR science arrays are rejected because they do not meet size thresholds.

For each granule, 2 HDF5 files are produced (SCRIF* and GCRSO* files). To improve processing latency, upfront calculations that were previously done for every granule, are now done only once for each processor core (4 for our test case). If the CrIS processing script runs normally, it will return a status code equal to zero. If the CrIS processing script encounters a fatal error, it will return a non-zero status code.

Note that the cross-granule requirements for the CrIS SDR are 4 "before" granules and 4 "after" granules. Therefore a sequence of 9 consecutive granules is required to process a single granule to SDR. For example, if the script is called on a single RDR file containing 15 consecutive granules, the middle 7 granules will be processed to SDRs. If the script is called on two RDR files containing 30 consecutive granules, the middle 22 granules will be processed.

To verify your output files against the output CrIS SDRs created at NOAA operations, execute the following commands:

```
cd ..
./cris_sdr_fsr_compare.bash idps_output_fsr_j01/ work_fsr_j01/
```

This script compares CSPP CrIS FSR output file arrays and Geolocation file Latitude and Longitudes with operational NOAA IDPS output files for all granules. The number of differences found will be printed. A successful test will result in few, if any differences.

6.2.2 Executing the CrIS NSR Test Case

To run the CrIS SDR NSR test case, unpack the test data as shown in Section 3.0 and then execute the commands below:

```
cd cris_test
mkdir work_nsr_j01
cd work_nsr_j01
cris_sdr.sh -n -p 4 ../input/RCRIS-RNSCA_j01*.h5
```

The test case executed using 4 processing cores should run in 1-2 minutes and create 7 output SDR granules. Note that to produce CrIS NSR resolution from NOAA-20 RDRs, the -n flag must be used. FSR is the default resolution for NOAA-20 CrIS, and NSR for S-NPP. The input RDR array sizes are scanned; in this test NOAA-20 test case, 2 spacecraft diary and 2 CrIS RDR science arrays are rejected because they do not meet size thresholds.

For each granule, 2 HDF5 files are produced (SCRIS* and GCRSO* files). To improve processing latency, upfront calculations that were previously done for every granule, are now done only once for each processor core (4 for our test case). If the CrIS processing script runs normally, it will return a status code equal to zero. If the CrIS processing script encounters a fatal error, it will return a non-zero status code.

Note that the cross-granule requirements for the CrIS SDR are 4 "before" granules and 4 "after" granules. Therefore a sequence of 9 consecutive granules is required to process a single granule to SDR. For example, if the script is called on a single RDR file containing 15 consecutive granules, the middle 7 granules will be processed to SDRs. If the script is called on two RDR files containing 30 consecutive granules, the middle 22 granules will be processed.

To verify your output files against the output CrIS SDRs created at NOAA operations, execute the following commands:

```
cd ..
./cris_sdr_nsr_compare.bash idps_output_nsr_j01/ work_nsr_j01/
```

This script compares CSPP CrIS FSR output file arrays and Geolocation file Latitude and Longitudes with operational NOAA IDPS output files for all granules. The number of differences found will be printed. A successful test will result in few, if any differences.

6.3 Creating CrIS Quicklook Images

To create quicklook images of the CrIS NSR or FSR shortwave, midwave and longwave infrared band brightness temperatures, use the ql_sdr.sh script:

```
$CSPP_SDR_HOME/viirs/ql_sdr.sh path_to_sdrs/
```

This script will produce images of brightness temperatures in the infrared longwave, midwave and shortwave regions using the SCRIS*.h5 or SCRIF*.h5 radiance and GCRSO*.h5 geolocation files in the `path_to_sdrs/` directory.

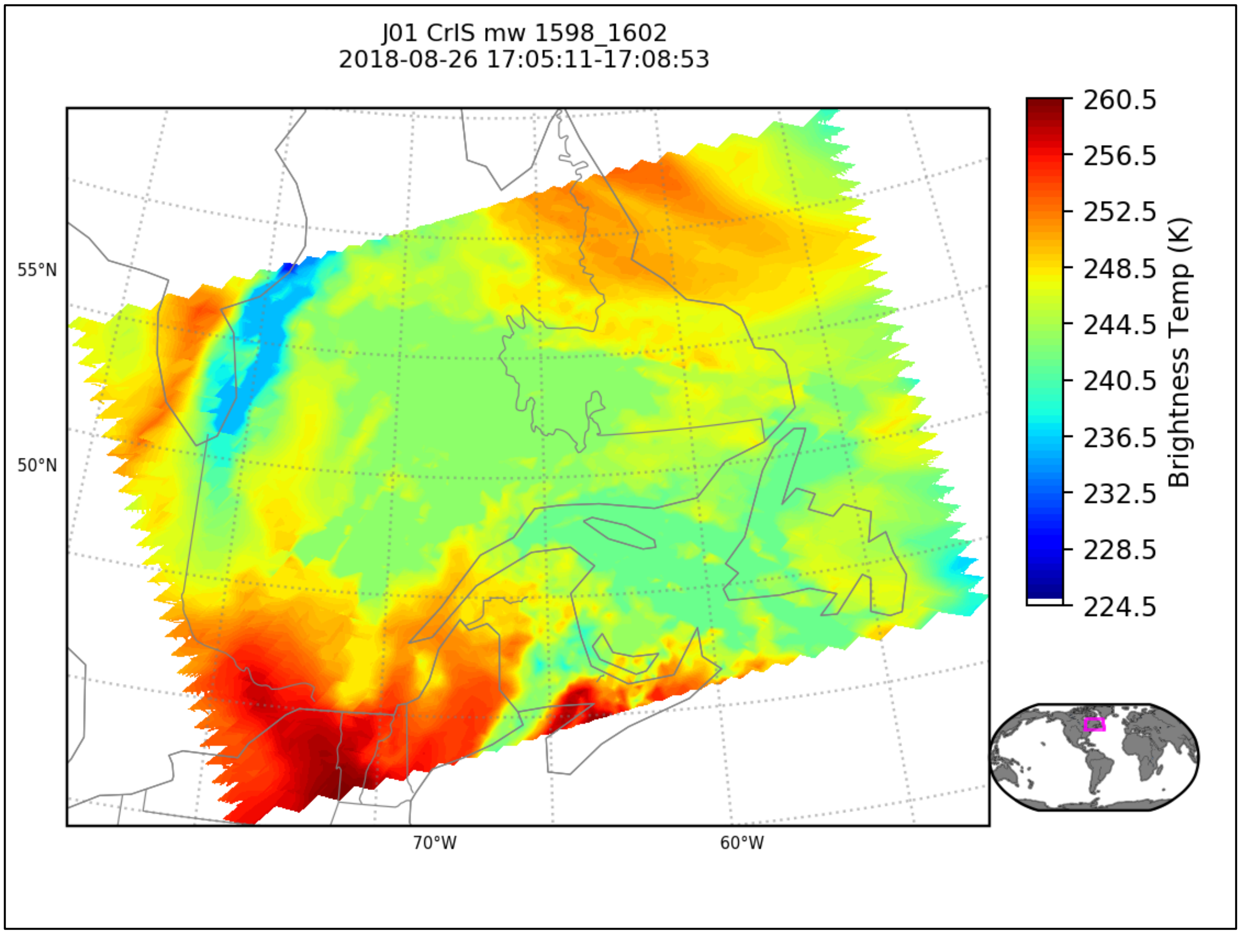
The output files will be created in the execution directory, e.g.

```
ql_sdr.sh work_fsr_j01/
```

will create brightness temperature images from files located in the `work_fsr_j01/` directory and place them in the current directory. An example of the quicklook CrIS infrared midwave image

SCRIF_2_j01_d20180826_t1705119_e1705417_b03992_c20180913171326805267_cspp_dev.png

created from the test data is shown below. Please note that the quicklook software interpolates through the two missing Fields-of- View (FOV) in each S-NPP CrIS 9x9 Field-of-Regard (FOR) that are excluded from the CrIS direct broadcast downlink for Suomi-NPP.



Other quick look options are available. Use ql_sdr.sh -h for a complete list.

Appendix 1: Frequently Asked Questions

A1.1 What should I do if I have a problem with CSPP SDR software?

If you have any problems with executing the CSPP SDR software, we ask that you follow these steps:

1. Make sure that you read the documentation and follow the instructions as written.
2. Run the verification tests to ensure a successful installation.
3. If you are still having problems, please send an email to cspp_issues@ssec.wisc.edu with a detailed description of the issue. Please include the following information,

    a. Exact CSPP SDR command you are using.
    b. Execution computer hardware and operating system.
    c. Computer memory.
    d. If the CSPP SDR execution failed, please send any log files that remain.

A1.2 How Can I Execute the Software without an Internet Connection?

Dynamic ancillary and Look-Up-Tables (LUTs) are required in order to create SDRs using the CSPP software. Therefore, it is necessary to have an internet connection in order to download the required files. Some users have requested instructions for use on processing machines that are behind firewalls that prevent outside access. To do this, you must have more than one instance of CSPP SDR software installed. One inside a firewall, and one used only to acquire the dynamic data.

1. On the machine without an internet connection, follow the instructions to do a full CSPP SDR Version 3.1 installation through Section 3.1. Do not execute the sdr_luts.sh or sdr_ancillary.sh scripts yet.
2. On the machine with an internet connection, do a partial installation of the CSPP SDR Version 3.1 software for downloading ancillary and auxiliary files

    ```
    mkdir CSPP
    cd CSPP
    tar xf ../CSPP_SDR_V3.1.tar.gz
    ```

3. Set the CSPP_SDR_HOME variable and source the environmental variables script on this connected machine, and then execute the sdr and ancillary update scripts as described in Sections 3.1 and 3.2, but make sure to include the -s option, which will download the files, but not place them in the local database. For example,

    ```
    sdr_luts.sh -s
    sdr_ancillary.sh -s
    ```

4. Now copy or rsync the new files from the machine with outside internet connection to the machine without it:

   ```
   rsync -av ${CSPP_SDR_HOME}/anc/cache/20*
   <machine2>:${CSPP_SDR_HOME}/anc/cache

   rsync -av ${CSPP_SDR_HOME}/anc/cache/incoming_luts/*
   <machine2>:${CSPP_SDR_HOME}/anc/cache/incoming_luts
   ```

5. Once this is complete, update the database on the machine without external internet:

   ```
   sdr_luts.sh -l
   sdr_ancillary.sh -l
   ```

6. From now on, when you execute the SDR driver scripts (viir_sdr.sh, cris_sdr.sh or atms_sdr.sh) on your processing machine, use the -l option so that the software will not attempt to download external files. For example,

   ```
   viirs_sdr.sh -p 4 -l ../rdr/*.h5
   ```

7. Set up routine download and copy of CSPP SDR ancillary and LUT files. Ancillary files (`sdr_ancillary.sh`) should be downloaded and transferred to the operational machine daily, while the LUTs should be checked weekly (`sdr_luts.sh`).

Appendix 2: Converting SDRs to BUFR Format for NWP Assimilation

A2.1 Script to convert ATMS SDR to BUFR (atms_bufr.bash) (5 November 2018)

```
#!/bin/bash

# This script converts all ATMS SDR files in the current directory to BUFR using AAPP.
# AAPP can be downloaded from https://www.nwpsaf.eu/site/software/aapp/download/
# For more information see the WMO Guide to DBNet at
# http://www.wmo.int/pages/prog/sat/documents/DBNet_Guide-to-DBNet.pdf
# Liam.Gumley@ssec.wisc.edu

# Set up AAPP environment
export AAPP_HOME=$HOME/nwpsaf/install      # Change this line for your AAPP install
export AAPP_PREFIX=$AAPP_HOME/AAPP_8.2
source $AAPP_PREFIX/ATOVS_CONF
export LD_LIBRARY_PATH=$AAPP_PREFIX/AAPP/lib:$LD_LIBRARY_PATH
export BUFR_TABLES=$AAPP_HOME/bufrdc/bufrtables/
export PATH=$AAPP_PREFIX/AAPP/bin:$PATH

# Set BUFR environment variables
export ORIGINATING_CENTRE=176              # Change this line for your DBNet center
export SUB_CENTRE=30                       # Change this line for your DBNet site
export USE_OB_TIME=Y

# Get list of SDR files
sdr=(SATMS*.h5)
tdr=(TATMS*.h5)
geo=(GATMO*.h5)
num=$(ls -1 SATMS*.h5 | wc -w)

# Create combined L1C file (atms.l1c)
for ((i = 0; i < num; i++)); do
  atms_sdr -g ${geo[$i]} ${sdr[$i]} ${tdr[$i]}
done
combine_1c SATMS*.l1c

# Create BUFR file (atms.bufr)
eccodes_encodebufr_1c -i atms.l1c ATMS
```

A2.1 Script to convert CrIS SDR to BUFR (cris_bufr.bash) (5 November 2018)

```
#!/bin/bash

# This script converts all CrIS SDR files in the current directory to BUFR using AAPP.
# AAPP can be downloaded from https://www.nwpsaf.eu/site/software/aapp/download/
# For more information see the WMO Guide to DBNet at
# http://www.wmo.int/pages/prog/sat/documents/DBNet_Guide-to-DBNet.pdf
# Liam.Gumley@ssec.wisc.edu

# Set up AAPP environment
export AAPP_HOME=$HOME/nwpsaf/install      # Change this line for your AAPP install
export AAPP_PREFIX=$AAPP_HOME/AAPP_8.2
source $AAPP_PREFIX/ATOVS_CONF
export LD_LIBRARY_PATH=$AAPP_PREFIX/AAPP/lib:$LD_LIBRARY_PATH
export BUFR_TABLES=$AAPP_HOME/bufrdc/bufrtables/
export PATH=$AAPP_PREFIX/AAPP/bin:$PATH

# Set BUFR environment variables
export ORIGINATING_CENTRE=176            # Change this line for your DBNet center
export SUB_CENTRE=30                     # Change this line for your DBNet site
export USE_OB_TIME=Y

# Get list of SDR and GEO files
sdr=(SCRI*.h5)
geo=(GCRSO*.h5)
num=$(ls -1 SCRI*.h5 | wc -w)

# Get file type for BUFR conversion (SCRIS or SCRIF)
file_type=$(basename ${sdr[0]} | cut -d"_" -f1)
if [ $file_type == "SCRIS" ]; then
  data_type="CRIS"       # CrIS NSR
else
  data_type="CRISFSR"    # CrIS FSR
fi

# Create combined L1C file (cris.l1c)
for ((i = 0; i < num; i++)); do
  cris_sdr -g ${geo[$i]} ${sdr[$i]}
done
combine_1c SCRI*.l1c

# Select CrIS channels
cris_channels cris.l1c cris_selection.l1c

# Create BUFR file (cris.bufr)
eccodes_encodebufr_1c -i cris_selection.l1c $data_type
```
