

Installation Instructions for the Community Satellite Processing Package for Geostationary Data (CSPP Geo) GRB Version 0.1 Prototype Software for GOES Rebroadcast

University of Wisconsin-Madison, Space Science and Engineering Center (SSEC)
Supported by the GOES-R Program Office and NOAA STAR

March 10 2015

Section 1: Introduction

1.1 Overview

This document contains instructions for installation and operation of the CSPP Geo prototype software package for ingesting baseband GOES Rebroadcast (GRB) data and processing to NetCDF output products.

The software in this release is a pre-launch prototype, capable of recovering and reconstructing ABI Level 1 and GLM Level 2 products from the GRB data stream. Future software releases will add the ability to recover data from other GOES-R instruments.

This package contains a mix of original software developed at the University of Wisconsin, third-party software libraries, and a modified version of RT-STPS, which is developed and maintained by the NASA Direct Readout Laboratory.

The CSPP Geo GRB Prototype package is distributed through the CSPP Geo website at:

<http://cimss.ssec.wisc.edu/csppgeo/>

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

1.2 System requirements

Minimum system requirements for the CSPP GRB software are as follows:

- 12 core, 2.4 GHz CPU with 64-bit instruction support,
- 32GB RAM,
- CentOS 6 64-bit Linux (or other compatible 64-bit Linux distribution),
- 100 GB disk space.

1.3 Test data

The test data is provided solely for purposes of testing the CSPP Geo GRB software, and is not intended to be used for product evaluation or verification.

The ABI test dataset was derived from output from an early GOES-R Ground System Data Operations Exercise (DOE-0). The contents are the ABI Triplet dataset, which was produced by the GOES-R Proxy Team. For more information on the ABI Triplet dataset, see

http://cimss.ssec.wisc.edu/goes_r/awg/proxy/nwp/ncsa_fcm/index.html

The GLM test dataset was created by running a simple test pattern through the Harris GRB simulator. The data is only useful for exercising the CSPP Geo software, and to provide an example of the L2 GLM data format. Because the data is entirely artificial, users should not expect it to be suitable for any other purpose, such as creating higher level products.

1.4 Disclaimer

Original source code, scripts and automation included as part of this package are distributed under the GNU GENERAL PUBLIC LICENSE agreement version 3. Binary executable files and third-party source code 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: CSPP Geo GRB Processing Modes

2.1 Processing Modes

The GRB software can be run in either of two processing modes, “streaming mode” or “command-line mode”. The following sections describe these two modes, including input data requirements.

In both modes, output is written to NetCDF files in the format described in the GOES-R Product Definition and User’s Guide (PUG), Rev C-1. The PUG is available from the GOES-R project website, although Rev C-1 has not yet been posted at the time of this release.

<http://www.goes-r.gov/resources/docs.html>

For more information on the software interface, refer to the “CSPP Geo GRB Software Version 1 Interface Control Document, Draft 20141114” (ICD), available on the CSPP Geo website. However, note the differences between the draft ICD and the interface implemented by this software version, as described in the following sections.

2.2 CSPP Geo GRB in Streaming Mode

In streaming mode, the GRB software runs as a daemon, reading a GRB data stream over a single socket and writing output to NetCDF. For this software version, input must be in the form of UDP datagrams containing Consultative Committee for Space Data Systems (CCSDS) packets wrapped with IPDU headers. For more information, see Appendix B: Creating input data for streaming mode.

Note that the streaming mode interface is slightly different from the planned operational interface, which is described in the draft ICD. For this release the streaming data format was simplified due to the lack of test data in the CADU-based format that is described in the ICD.

In addition, a future release will add support for reading data from two sockets to simultaneously process right-hand and left-hand polarization streams, as described in the ICD.

The main control script for extracting CCSDS packets from the GRB data stream is `cspg-rt-grb.sh`. This script uses the Java Service Wrapper to start a GRB ingestor. The service wrapper will monitor the ingestor and restart in the event of any failure.

The GRB ingestor will watch for UDP traffic on port 50002. It extracts CCSDS packets, bundles them in intermediate output files by packet count or elapsed time according to your site configuration (see Appendix D: Streaming Mode

Configuration Settings), with a default bundling of one file written every 15 seconds. These bundles will be written to the location specified by the environment variable `CSPP_GEO_RTCSP_OUT`.

The packet bundle file naming convention is described in Appendix A: Intermediate CCSDS file naming convention.

After each packet bundle is written, the GRB Reconstruction software (see “User-Callable Mode”) is automatically executed to process the GRB space packets, assembling the data payloads into NetCDF output files. These NetCDF files will be written to the location specified by the environment variable `CSPP_GEO_GRBR_OUT`.

2.3 CSPP Geo GRB in Command-Line Mode

In command-line mode, the software is called at the command line, and processes files containing CCSDS packets. Input files must contain only unsegmented packets (with APID 0b11) or complete sequences of packets.

A complete sequence of packets has the following characteristics:

- a single, identical APID,
- begins with a packet containing primary header sequence flag 0b01,
- ends with a packet containing primary header sequence flag 0b10,
- incrementing sequence count (mod 16384),
- no missing sequence counts.

To see an example of input files, run the GRB software in streaming mode, and have a look at the intermediate files left in the location specified by the environment variable `CSPP_GEO_RTCSP_OUT`.

To run in command-line mode, provide any number of CCSDS packet bundles to the GRB Reconstruction script:

```
cspp-grbr.sh filename [filename ...]
```

Output can be found in the location specified by the environment variable `CSPP_GEO_GRBR_OUT`.

Section 3: Installation and Running the Test Case

3.1.1 System Configuration

Given the high data rate for the GRB signal, it is recommended the system memory buffers for UDP datagram traffic be increased from their default settings. This will provide more flexibility to keep up with the data rate in the event of short-lived fluctuations on system resources demands. Our suggested settings are below; you will likely need root access on your machine to make these changes.

```
echo 16777216 > /proc/sys/net/core/rmem_max
echo 16777216 > /proc/sys/net/core/wmem_max
echo 16777216 > /proc/sys/net/core/rmem_default
echo 16777216 > /proc/sys/net/core/wmem_default
```

3.1.2 Installation of GRB software

Download the following file from the CSPP website

<http://download.ssec.wisc.edu/files/csppgeo/> :

```
cspp-geo-grb-0.1.tar.gz
```

Install the software as shown below (a new directory named cspp-geo-grb-0.1 will be created). In this example, the tar files are assumed to be in the user's home directory.

```
tar xzf cspp-geo-grb-0.1.tar.gz
```

Set the `CSPP_GEO_GRB_HOME` environment variable to the name of the directory where CSPP Geo GRB was installed (`$HOME/cspp-geo-grb-0.1` in this example), and then execute the environment setup script as shown below:

```
export CSPP_GEO_GRB_HOME=$HOME/cspp-geo-grb-0.1
source $CSPP_GEO_GRB_HOME/grb_env.sh
```

These lines can be added to your shell login script if you plan to run the GRB software again.

The default locations of environment variables, such as `CSPP_GEO_GRB_OUT` can be modified in the `grb_env.sh` script. For the rest of this document, we assume you have set up your environment accordingly.

In order to run this test case, port 50002 must be free. Information on additional system configuration changes are mentioned in Section 4.1, but are not required for this example.

To check the availability of port 50002, run the following command:

```
lsof -i -n|grep 50002
```

If the port is free, the command should not return any output. If you encounter output, please check with your system administrator before proceeding.

To start the GRB ingestor running:

```
cspp-rt-grb.sh start
```

To verify correct startup and data processing, you can tail the log file:

```
tail -f $CSPP_GEO_RTCSP_HOME/jsw/logs/grb.log
```

In the log, you should see information similar to:

```
INFO | jvm 1 | 2015/03/05 18:22:21 | 301
INFO | jvm 1 | 2015/03/05 18:22:21 | 302
INFO | jvm 1 | 2015/03/05 18:22:21 | 303
INFO | jvm 1 | 2015/03/05 18:22:21 | 304
INFO | jvm 1 | 2015/03/05 18:22:21 | Start time:
1425579741487
```

The start time in the log indicates that the GRB ingestor started successfully and is ready for testing.

To stop the GRB ingestor:

```
cspp-rt-grb.sh stop
```

3.2 Running the GRB Test Case

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

```
cspp-geo-grb-test-data-0.1.tar.gz
```

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

```
cd $HOME
tar xzf cspp-geo-grb-test-data-0.1.tar.gz
```

This test case will simulate the CSPP Geo GRB in streaming mode by running a data-feeder test script. This test script will take the sample DOE test data, convert it to UDP datagrams containing CCSDS packets wrapped with IPDU headers, and send these to the port RT-CSPP expects data on, simulating a GRB data stream.

First, start the GRB ingestor:

```
cspp-rt-grb.sh start
```

The GRB ingestor is now up and waiting for data.

Now change to the testing directory:

```
cd $HOME/cspp-geo-grb-test-data-0.1
```

Run the data-feeder test script.

```
./test-grb.sh
```

If the test ran successfully, you will see CCSDS GRB packet bundles in the RT-CSPP output directory (`$CSPP_GEO_GRB_HOME/output/RT-CSPP`), similar to:

```
GRB_250_131_2015-04-06T16:44:19.983GMT.ccsds  
GRB_250_304_2015-04-06T15:38:17.375GMT.ccsds  
GRB_250_132_2015-04-06T16:44:31.956GMT.ccsds
```

...etc., each file containing a site-configurable number of packets.

You should also find reassembled NetCDF product files in the GRB-R output directory (`$CSPP_GEO_GRB_HOME/output/GRB-R`), similar to:

```
ABI-L1b-RadM2-M3C13_G16_s2016033120000.nc
```

When the test is complete, go back to your GRB base directory and stop the GRB ingestor:

```
cspp-rt-grb.sh stop
```

Note that, for this prototype, output clean-up is left to the user. Reconstructed NetCDF files are named based on the data, so it is recommended that you remove any files in `$CSPP_GEO_GRBR_OUT` before re-running any test cases to avoid unexpected behavior when the software tries to reconstruct a file that already exists.

Also note that NetCDF reconstructor processes can remain running for up to five minutes after the test script has stopped sending data.

3.3 Creating GRB Quicklook Images

After the GRB Test Case has been successfully run, you may generate PNG quicklook images for the NetCDF files that were created by using the `cspp-ql-grb.sh` script:

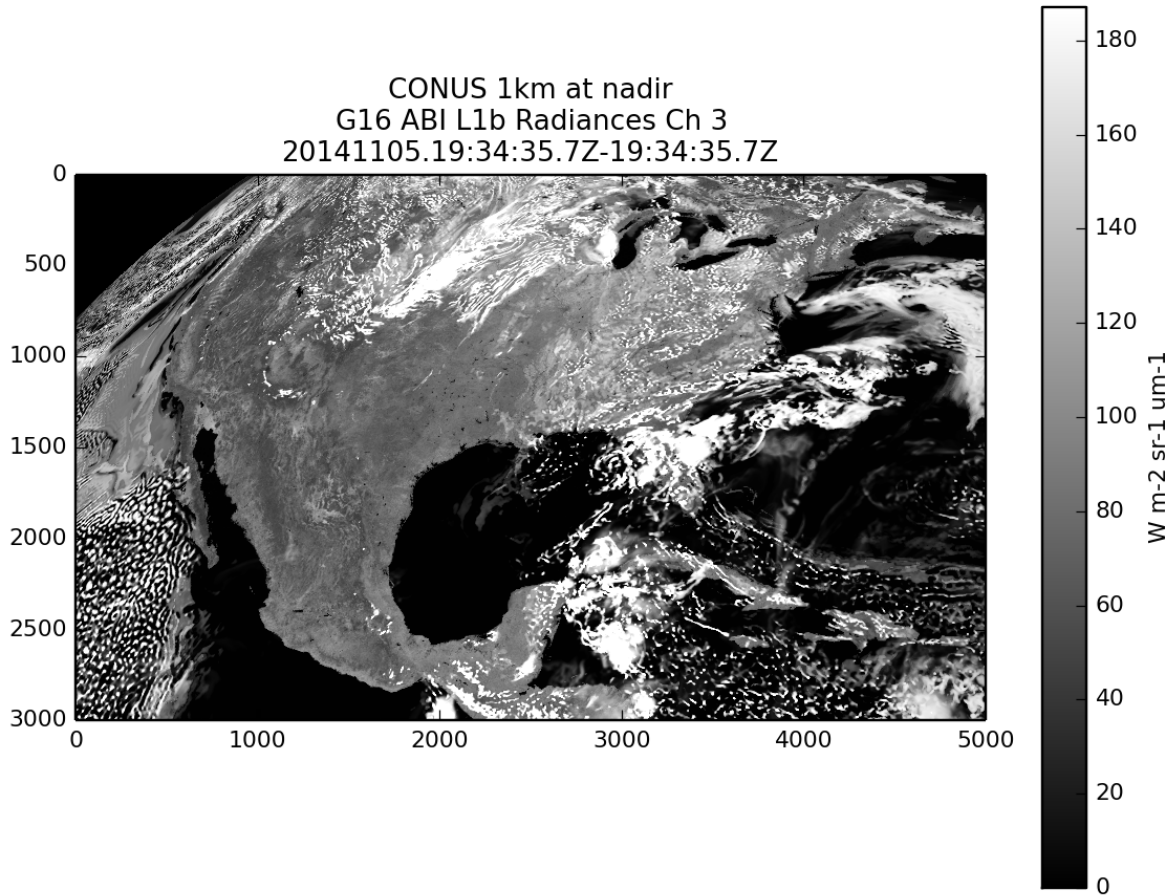
```
cspp-ql-grb.sh file_name(s)
```

For example, run the following quicklook command:

```
cspp-ql-grb.sh $CSPP_GEO_GRBR_OUT/ABI*.nc
```

This will create images of all the ABI NetCDF files in the GRB output directory. **Note:** The quicklook command will create over 500 quicklook images from the provided test data and may take longer than 15 minutes to complete.

An example of the derived ABI L1b Radiances for Channel 3 for 20141105 at 19:34:35Z is shown below:



Section 4: Advanced Configuration

4.1. System Configuration

To accommodate processing of the 31 Mb/s GRB data stream, there are a few configuration changes that will need to be made by a system administrator (requiring root access). While all of these changes were not needed to run the test case provided with the GRB software, they may be needed to process data from an external device such as a demodulator or GRB simulator.

The CSPP Geo GRB ingest software receives data on UDP port 50002 through the Ethernet interface which is connected to your GRB data source. If a local firewall is running on the machine that is ingesting data, be sure that the firewall allows traffic on those ports through the connected interface.

Activate an interface to the IP address acting as the source of your data, such as a demodulator or GRB simulator.

```
ifconfig <interface> <IP address of demod>
```

Section 5: Known Issues

5.1 Caveats and known issues

1. The NetCDF output file naming convention is likely to change in a future release, during a finalization step that cleans up the output and renames the file to match the `dataset_name` attribute in the metadata. Currently, no cleanup of intermediate files is performed for this prototype release.
2. There is a known omission in the `ql_grb.py` quicklook generation code which can cause graphical back-end errors (*“ImportError: No module named sip”*) when the quicklooks are run. To fix this, insert the line `import matplotlib; matplotlib.use('Agg')` before `import logging` in `ql_grb.py`. This will be fixed in a future beta.

Appendix A: Intermediate CCSDS file naming convention

This appendix describes the naming convention of the intermediate CCSDS files that are written by the ingestor when running in streaming mode. Note that this is simply the convention used by the ingestor, and is not a requirement for processing in command-line mode.

The file naming pattern is:

GRB_SID_APD_YYYY-MM-DDThh:mm:ss.sssTZD.ccsds

This file naming pattern represents:

GRB: Data source, always GRB
SID: Spacecraft ID: always 250
APD: The hexadecimal APID this data bundle contains (reference PUG Table A.1)
YYYY-MM-DD: Complete date, e.g. 2013-07-16
hh:mm:ss.sss: Hours, minutes, seconds, and milliseconds
TZD: Time zone designator
ccsds: suffix, always ccsds

A sample packet bundle filename might be:

GRB_250_303_2015-03-25T07:32:03.722GMT.ccsds

Appendix B: Creating input data for streaming mode

A file containing GRB CCSDS packets (i.e. a “CCSDS bundle”) can be run through the GRB software in streaming mode by first converting it into the format of the DOE NetCDF GRB files, then running it through a sender script.

First, run the script `$CSPP_GEO_GRB_HOME/GRB-R/src/ccsds2nc.py` to convert one or more CCSDS bundles to a NetCDF file:

```
$CSPP_GEO_GRB_HOME/GRB-R/src/ccsds2nc.py output.nc ccsds_file  
[ccsds_file ...]
```

To process the data, run the script `sender.py`. Refer to `test-grb.sh` as an example for how to do this. The sender will insert 32-byte IPDU headers in between the packets and stream via UDP to the GRB software. Note that the IPDU headers do not need to contain real values, since they are skipped by the GRB software.

Appendix C: Validating input data for command-line mode

To validate CCSDS packet bundles for compliance with the PUG, run:

```
$CSPP_GEO_GRB_HOME/GRB-R/src/validator.py -v bundle.ccsds
```

If any issues are found in the data, errors or warnings will be printed to screen. If there is no output, no problems were found. Note that not all errors are fatal for GRB processing.

The validator currently only checks the packet headers for valid values, but will become more full-featured in future releases.

Example 1: In this case, assembler ID 12 is not consistent with the PUG, however GRB processing will not be affected

```
$CSPP_GEO_GRB_HOME/GRB-R/src/validator.py -v \  
GRB_250_137_2015-02-06T11:36:54.239GMT.ccsds  
WARNING: __main__:Unrecognized assembler_id: 12  
ERROR: __main__:1 errors found in packet @ offset 0  
WARNING: __main__:Unrecognized assembler_id: 12  
ERROR: __main__:1 errors found in packet @ offset 16390
```

Example 2: In this case, the packet payload headers are missing valid times. This could cause GRB processing to overwrite files.

```
$CSPP_GEO_GRB_HOME/GRB-R/src/validator.py -v \  
GRB_250_300_2015-06-04T23:00:00.327GMT.ccsds  
WARNING: __main__:Invalid (<= 0) seconds_since_epoch: 0  
WARNING: __main__:Invalid (<= 0) microseconds_of_second: 0  
ERROR: __main__:2 errors found in packet @ offset 0
```

Appendix D: Streaming Mode Configuration Settings

Configuration changes for the GRB ingestor streaming mode can be made by editing the following file:

```
$CSPP_GEO_RTCSP_HOME/jsw/conf/grb-udpserver.conf
```

The parameters of interest to most sites will be found in the Java Additional Parameters section, and include the two parameters which dictate how often a CCSDS packet bundle for a particular APID is written to disk:

```
wrapper.java.additional.5=-DMAX_PACKETS_PER_BUNDLE=1000  
wrapper.java.additional.6=-DMAX_ELAPSED_TIME_PER_BUNDLE=30
```

These two parameters constrain how often the ingestor writes a packet bundle; in this case, either every 1,000 packets, or every 30 seconds, whichever occurs first for each selected APID. The APIDs selected to be kept and stored by the ingestor are specified by:

```
wrapper.java.additional.7=-DAPID_MASK=100-19F,300-304
```

The mask is a comma-separated list of APID ranges, in hexadecimal format. The above mask represents all possible ABI and GLM APIDs.

Most other configuration parameters should remain at the default values. For example, the IP address and network ports are set to default values for the GRB Simulator. For questions on any other parameters, contact SSEC via the CSPP Geo web site.