Installation Instructions for the Community Satellite Processing Package for Geostationary Data (CSPP Geo) GRB Version 0.3 Prototype Software for GOES Rebroadcast

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

October 6 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-4 output products.

The software in this release is capable of recovering and reconstructing Advanced Baseline Imager (ABI) Level 1b, Geostationary Lightning Mapper (GLM) Level 2+, Space Environment In-Situ Suite (SEISS) Level 1b, Magnetometer (MAG) Level 1b, Solar Ultraviolet Imager (SUVI) Level 1b, and Extreme Ultraviolet and X-ray Irradiance Sensors (EXIS) Level 1b products from the GRB data stream.

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' form on the website to submit any questions or comments about CSPP.

This software was developed according to the GRB specifications described in the GOES-R Product Definition and User's Guide (PUG), Vol. 4, Rev D. The PUG is available from the GOES-R project website, although Rev D has not yet been posted at the time of this release.

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

1.2 What's New in CSPP Geo GRB Version 0.3

Changes for CSPP Geo GRB Version 0.3 include the following:

- Added support for the EXIS and SUVI instruments.
- Added more configuration options for output, see grb_env.sh for details.
- Added support for running from a read-only location if properly configured.
- Selecting Application Identifiers (APIDs) to capture is now done in GRB.xml, APID_MASK has been removed.
- Various tuning and performance tweaks to ingestor and reconstructor

1.3 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.

1.4 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 test dataset was derived from output from a GOES-R Ground System Data Operations Exercise (DOE-1/2). The contents include 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

1.5 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-4 files in the format described in the PUG.

For more information on the software interface, refer to the "CSPP Geo GRB Software Version 0.3 Interface Control Document, 20150917" (ICD), available on the CSPP Geo web site.

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-4. Input must be in the form of UDP datagrams containing Consultative Committee for Space Data Systems (CCSDS) Channel Access Data Unit (CADU) frames, each exactly 2048 bytes in length. For more information, see Appendix A: Creating input data for streaming mode.

Data can be read 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 cspp-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 simultaneously on two network ports. It extracts CADU frames, assembles the CADUs into CCSDS packets, filters unwanted (e.g. idle) packets, and bundles desired packets in intermediate output files by packet count or elapsed time according to your site configuration (see Appendix B: Streaming Mode Configuration Settings), with a default bundling of one file written every 30 seconds. These bundles will be written to the location specified by the environment variable `CSPP_GEO_RTCSPP_OUT`.

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

After each packet bundle is written, the GRB Reconstruction software (see "Command-Line Mode") is automatically executed to process the GRB space packets, assembling the data payloads into NetCDF-4 files. These files will be written to the location specified by the environment variable `CSPP_GEO_GRBR_OUT`. SUVI Flexible Image Transport System (FITS) files can currently be created manually. See Appendix D: Creating FITS Files for more information.

2.3 CSPP Geo GRB in Command-Line Mode

In command-line mode, the software is explicitly called by the user at the command line, and processes binary files containing CCSDS packets. Refer to the ICD for details on input requirements.

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_RTCSPP_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 maximum network send and receive buffer sizes 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. To check the values of the buffer sizes:

```
cat /proc/sys/net/core/rmem_max
cat /proc/sys/net/core/wmem_max
```

CSPP Geo GRB will not run if the values do not meet or exceed 16777216. 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
```

In order to retain these settings between system reboots, placing both lines at the end of `/etc/rc.local` is one way to configure kernel parameters in Linux. This

should be done in addition to executing the two lines interactively during initial installation and testing.

3.1.2 Installation of GRB software

Download the following file from the CSPP Geo web site,
http://download.ssec.wisc.edu/files/csppgeo/ :

```
cspp-geo-grb-0.3.tar.gz
```

Install the software as shown below (a new directory named cspp-geo-grb-0.3 will be created). In this example, the tar files are assumed to be in the user's home directory.

```
tar xzf cspp-geo-grb-0.3.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-0.3` in this example), and then execute the environment setup script as shown below:

```
export CSPP_GEO_GRB_HOME=$HOME/cspp-geo-grb-0.3
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 settings 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.

To start the GRB ingestor running:

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

This command starts up a daemon process which runs in the background, listening for GRB data on ports 5530 and 5531. To verify correct startup and data processing, you can tail the log file:

```
tail -n 15 -f $CSPP_GEO_GRB_HOME/output/var/log/grb.log
```

In the log, you should see information similar to:

```
INFO | jvm 1  | 2015/09/02 19:31:13 | APID to process: 1162
INFO | jvm 1  | 2015/09/02 19:31:13 | APID to process: 1163
INFO | jvm 1  | 2015/09/02 19:31:13 | APID to process: 1280
INFO | jvm 1  | 2015/09/02 19:31:13 | APID to process: 1281
INFO | jvm 1  | 2015/09/02 19:31:13 | APID to process: 1408
INFO | jvm 1  | 2015/09/02 19:31:13 | Link Name: grb-bundle
INFO | jvm 1  | 2015/09/02 19:31:13 | finishSetup() in...
```

This section of the log shows the tail end of the list of APIDs the ingestor will process, and indicates the final node in the processing chain, which writes GRB packet bundles to disk.

If the GRB ingestor was not able to start, the last line of the log will look this this:

```
STATUS |wrapper| 2015/08/21 23:24:04 | <-- Wrapper Stopped
```

The likely cause of the error is either insufficient maximum network buffer sizes, or the ports required by the GRB ingestor are not available. Refer to Section 3.1.1 to check the maximum network buffer sizes. To check the availability of ports 5530 and 5531, run the following commands (root access required):

```
sudo lsof -i -n | grep 5530
sudo lsof -i -n | grep 5531
```

If the ports are free, the command should not return any output. If you encounter output, you can resolve the problem by shutting down or reconfiguring the program that is using the ports, or by configuring the GRB ingestor to use different ports (see Appendix B). Then start the ingestor as described above.

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-0.3.tar.gz
```

This tarball contains GRB test data for all six supported instruments in the form of CCSDS packets, as well as a data feeder test script. The test script frames the packets into CADUs, then sends the data as UDP datagrams to the ports monitored by the CSPP Geo GRB software, simulating a GRB data stream as delivered by a demodulator.

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-0.3.tar.gz
```

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-0.3
```

Run the data-feeder test script.

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

The test should take about 7 minutes to run. You should see some screen output similar to:

```
Sending to 127.0.0.1:5530 (LHCP) and 127.0.0.1:5531 (RHCP)
Start at: 2015-09-09 15:49:44.708745
Sent 100 of 100 files; 89986 packets; 564835 cadus; 1148877064
bytes..
Done at: 2015-09-09 15:54:51.076428
Effective bytes per second: 3749994.29689
Highest single burst: 2177245
Test complete. Waiting for end-of-stream flush...
```

To confirm the test ran successfully, compare the number of packets sent (89986) with the number of packets processed by running this command:

```
grep "Num packets"
$CSPP_GEO_GRB_HOME/output/var/log/grb.log|tail -n 1
```

If the test ran successfully, you will see output similar to:

```
INFO| jvm 1 | 2015/09/09 16:09:10 | Num packets seen: 89986
```

which matches the number of packets sent in the output above. In addition, there will be CCSDS GRB packet bundles in `$CSPP_GEO_RTCSPP_OUT` (default is `$CSPP_GEO_GRB_HOME/output/RT-CSPP`), similar to:

GRB_0_131_2015-04-06T16:44:19.983Z.ccsds
GRB_0_304_2015-04-06T15:38:17.375Z.ccsds
GRB_0_132_2015-04-06T16:44:31.956Z.ccsds
…etc., each file containing a site-configurable number of packets.

You should also find reassembled NetCDF-4 product files in `$CSPP_GEO_GRBR_OUT` (default is `$CSPP_GEO_GRB_HOME/output/GRB-R`), similar to:

ABI-L1b-RadM2-M3C13_G16_s2015167133301.nc
EXIS-EUV_G16_s2015167110434.nc
GLM_G16_s2015167133209.nc
MAG_G16_s2015167133301.nc
SEISS-EHI_G16_s2015167133200.nc
SUVI-Fe094_G16_s2015167134914.nc

When the test is complete, stop the GRB ingestor:

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

Note that, for this prototype release, output cleanup is left to the user. Reconstructed NetCDF-4 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-4 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 from the ABI NetCDF-4 files that were created by running the cspp-ql-grb.sh script. The general form of the invocation is:

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

To generate quicklook images for all ABI NetCDF-4 files, specify the file pattern with wildcards as shown below. Note this will result in over 600 images, and may take more than 10 minutes.

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

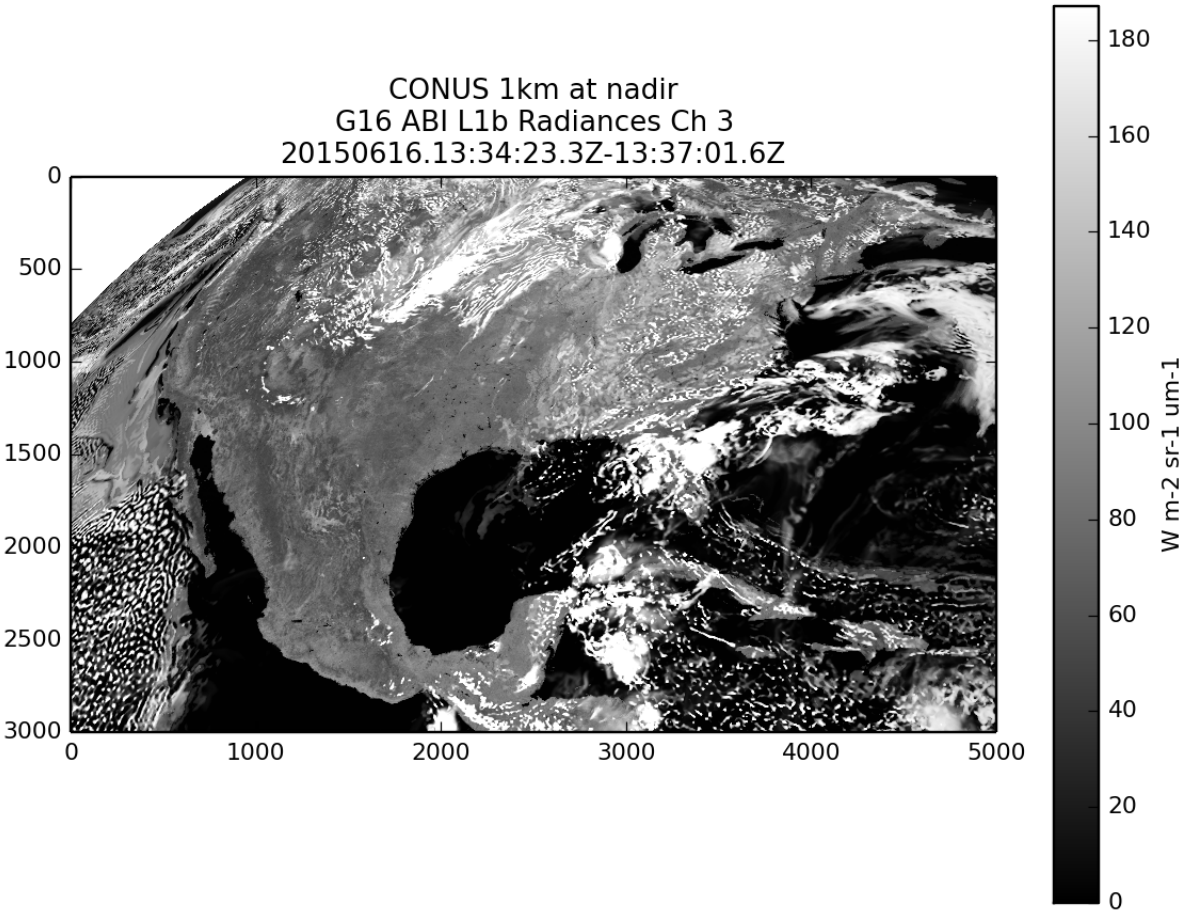
To generate quicklook images for only full disk or CONUS NetCDF-4 files, use one of the invocations below.

```
cspp-ql-grb.sh $CSPP_GEO_GRBR_OUT/ABI-L1b-RadF*.nc
cspp-ql-grb.sh $CSPP_GEO_GRBR_OUT/ABI-L1b-RadC*.nc
```

Quicklooks can be generated much more quickly by running multiple jobs in parallel, using the Linux “xargs” command. The following invocation generates the same quicklooks in just a few minutes, using 8 cores:

```
ls ${CSPP_GEO_GRBR_OUT}/ABI*.nc | xargs -t -n1 -P8 cspp-ql-grb.sh
```

An example quicklook image of the derived ABI CONUS L1b Radiances for Channel 3 for 20150616 at 13:34:23Z is shown on the next page:



If large rectangular white areas are visible in quicklooks generated from test case output, it is an indication that one or more ABI image packets were dropped during the test. Check that the system meets the minimum hardware requirements, that the system memory buffers were configured as previously described in this document, and that there are no other jobs consuming significant resources or other performance issues with the test machine when the test case is run.

Quicklook images for instruments other than ABI are not supported in this release.

Section 5: Known Issues

5.1 Caveats and known issues

1. The NetCDF-4 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. The GRB software currently uses a large number of processes; this will improve in future releases. If grb.log reports "`java.io.IOException: Cannot run program "../cspp-geo-grb-0.3/scripts/cspp-grbr.sh": error=11, Resource temporarily unavailable`", you have likely encountered the user process limit on your system, which could cause the software to miss packets. Many environments set this limit at 1024 processes but will allow a user to increase their own limit for a session via: `` `ulimit -u 3000` ``, where 3000 is a limit that we've found sufficient.

Appendix A: 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-4 GRB files, then running it through a sender script.

First, run the script 'ccsds2nc.py' to convert one or more CCSDS bundles to a NetCDF-4 file:

```
$CSPP_GEO_GRBR_HOME/ShellB3/bin/python \
$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.

Appendix B: Streaming Mode Configuration Settings

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

```
$CSPP_GEO_RTCSPP_HOME/jsw/conf/grb-ingestor.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.

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. If your site needs to change the default IP address and network ports, edit these three lines:

```
wrapper.java.additional.8=-DsourceIpAddress=localhost
wrapper.java.additional.9=-DleftPort=5530
wrapper.java.additional.10=-DrightPort=5531
```

To run the GRB Test Case as described in section 3.2, the `sender.py` script in the test data package will also require changes. Edit these three lines:

```
UDP_IP = '127.0.0.1'
UDP_LEFT_PORT = 5530 # port for left-hand polarization data
UDP_RIGHT_PORT = 5531 # port for right-hand polarization
data
```

To activate an interface to the IP address acting as the source of your data, such as a demodulator or GRB simulator, run the following command:

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

For questions on any other parameters, contact SSEC via the CSPP Geo web site.

Appendix C: 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.sssZ.ccsds
```

This file naming pattern represents:
`GRB`: Data source, always “GRB”
`SID`: Spacecraft ID: Currently zero, but expect 250 for live GOES-R data
`APD`: The hexadecimal APID this data bundle contains (reference PUG Vol. 4 Table A.1)

YYYY-MM-DD: Year, Month, and Day the packet bundle was written, e.g. 2013-07-16
T: Delimiter between date and time, <date>T<time>, as per ISO-8601 specification
hh:mm:ss.sss: Hours, minutes, seconds, and milliseconds
Z: Time zone designator, always "Z" indicating UTC, per ISO-8601 specification
ccsds: suffix, always "ccsds"

A sample packet bundle filename might be:

```
GRB_0_303_2015-03-25T07:32:03.722Z.ccsds
```

Appendix D: Creating FITS files

A utility is included to convert SUVI NetCDF-4 files to FITS format. To use it, run:

```
$CSPP_GEO_GRBR_HOME/ShellB3/bin/python \
$CSPP_GEO_GRBR_HOME/src/suvi-nc2fits.py input.nc
```

A .fits file will be created in the same directory as the input. The CSPP Geo team would like to solicit feedback from FITS users to help us ensure that these files work with their downstream software. Please use the "Contact Us" link on the CSPP Geo web site to submit any feedback.

Appendix E: Validating input data for command-line mode

To validate CCSDS packet bundles, run:

```
$CSPP_GEO_GRB_HOME/GRB-R/ShellB3/bin/python \
$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 Vol. 4, however GRB processing will not be affected

```
$CSPP_GEO_GRB_HOME/GRB-R/src/validator.py -v \
GRB_0_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_0_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 F: Configuring which data to process by Application Identifier (APID)

Which GRB products are created is controlled by an XML configuration file, located in:

```
$CSPP_GEO_RTCSPP_HOME/config/GRB.xml
```

By default, all APIDs defined in the PUG (reference PUG Vol. 4 Table A.1) are processed. If your site wishes to change the default settings (for example, if your site is only interested in ABI, and would prefer to filter out data for all other instruments), edit the XML file and comment out all occurrences of XML elements for the APIDs you wish to remove. The standard html comment tag of "<!-- ... -->" can be used. For example, to filter out data for the SEISS Solar and Galactic Protons, search for that term, and comment out each XML element for the metadata and data APIDs so it looks like:

```
<!-- 430 1072 SEISS Solar and Galactic Protons Metadata -->
<!-- <pklink appid="1072" label="apid1072" /> -->
<!-- 431 1073 SEISS Solar and Galactic Protons Data -->
<!-- <pklink appid=”1073” label=”apid1073” /> -->
```

Note, there are currently five references in GRB.xml for each APID (one for each node in the RT-CSPP processing pipeline). For each APID you want to stop processing, all five XML elements must be commented out. This may be simplified in a future release. For each APID, both the hexadecimal and decimal values are provided in the XML descriptions (in the above example hexadecimal 0x430, decimal 1072 for the metadata and 0x431, decimal 1073 for the data).