

Real-time Software Telemetry Processing System (RT-STPS) User's Guide

Version 7.0

November 2022



**GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND**

Table of Contents

1	General	1
2	Software Description	1
3	Software Version	2
4	Program Inputs and Outputs	3
5	Installation and Configuration	3
5.1	Installation	4
5.1.1	Requirements	4
5.1.2	Installation Instructions	4
5.1.3	Configure RT-STPS	5
5.1.4	leapsec File Updates	6
5.1.5	Package Layout	6
5.1.6	Testing the Installation	7
5.1.6.1	Start the Server Manually	7
5.1.6.2	Execute the Viewer	7
5.1.6.3	Execute the Sender	8
5.1.6.4	Inspect the Results	9
5.1.6.5	Stop the Server	10
5.1.7	Creating Launchers	10
5.1.8	Firewall Configuration	10
5.2	Java Setup and Performance Notes	10
6	Program Operation	10
6.1	Provided Configuration Files	11
6.2	Starting the Server	12
6.2.1	Command Line Script	12
6.2.2	Java Service Wrapper	12
6.3	Viewer	12
6.4	Sender	13
6.5	Batch Mode	13
6.6	Logging	13
6.7	Stopping the Server	14
6.7.1	Command Line Script	14
6.7.2	Java Service Wrapper	14
6.8	Automatic Setup	14
7	Raw Data Record (RDR) Creation	15
7.1	Supported RDRs	15
7.2	Test Data	15
7.3	Suomi NPP, JPSS-1, and JPSS-2 Configuration File for RDRs	15
7.4	Processing the Test Data	16
7.5	RDR Processing Status	16
7.6	Expected Output	16
7.7	Reference Sources	16
8	Additional Command Line Tools	17
8.1	Getstatus	17

8.2	Load	17
8.3	Shutdown	17
8.4	Version	17
8.5	Rate Buffering Program	17
9	Alternate Server Interface	18
Figure 1.	RT-STPS Architecture	2
Figure 2.	RT-STPS Viewer	8
Figure 3.	RT-STPS Sender	9
Figure 4.	RDR Creation Process	15

1 General

The NASA Goddard Space Flight Center's (GSFC) Direct Readout Laboratory (DRL) developed the Real-time Software Telemetry Processing System (RT-STPS) software.

Users must agree to all terms and conditions in the Software Usage Agreement on the DRL Web Portal before downloading this software.

Software and documentation published on the DRL Web Portal may occasionally be updated or modified. The most current versions of DRL software are available at the DRL Web Portal:

<https://directreadout.sci.gsfc.nasa.gov>

Questions relating to the contents or status of this software and its documentation should be addressed to the DRL via the Contact DRL mechanism at the DRL Web Portal:

<https://directreadout.sci.gsfc.nasa.gov/?id=dspContent&cid=66>

2 Software Description

RT-STPS ingests raw telemetry data and produces products, including sorted Consultative Committee for Space Data Systems (CCSDS) packets and Virtual Channel Data Units (VCDUs). RT-STPS supports a variety of output formats.

Installed as a server RT-STPS will operate continuously, receiving data from a port or a file and outputting results to files and sockets as specified in a configuration file. A separate interface can be used to invoke RT-STPS from the command line.

The RT-STPS package includes two main utilities: the viewer and the sender. The viewer displays the progress of the server as it runs, and it can be used to load server configuration files. The sender copies a raw data file to the server for processing.

The RT-STPS architecture is depicted in Figure 1. Each server component performs a typical part of the overall CCSDS processing from raw telemetry frames to packets.

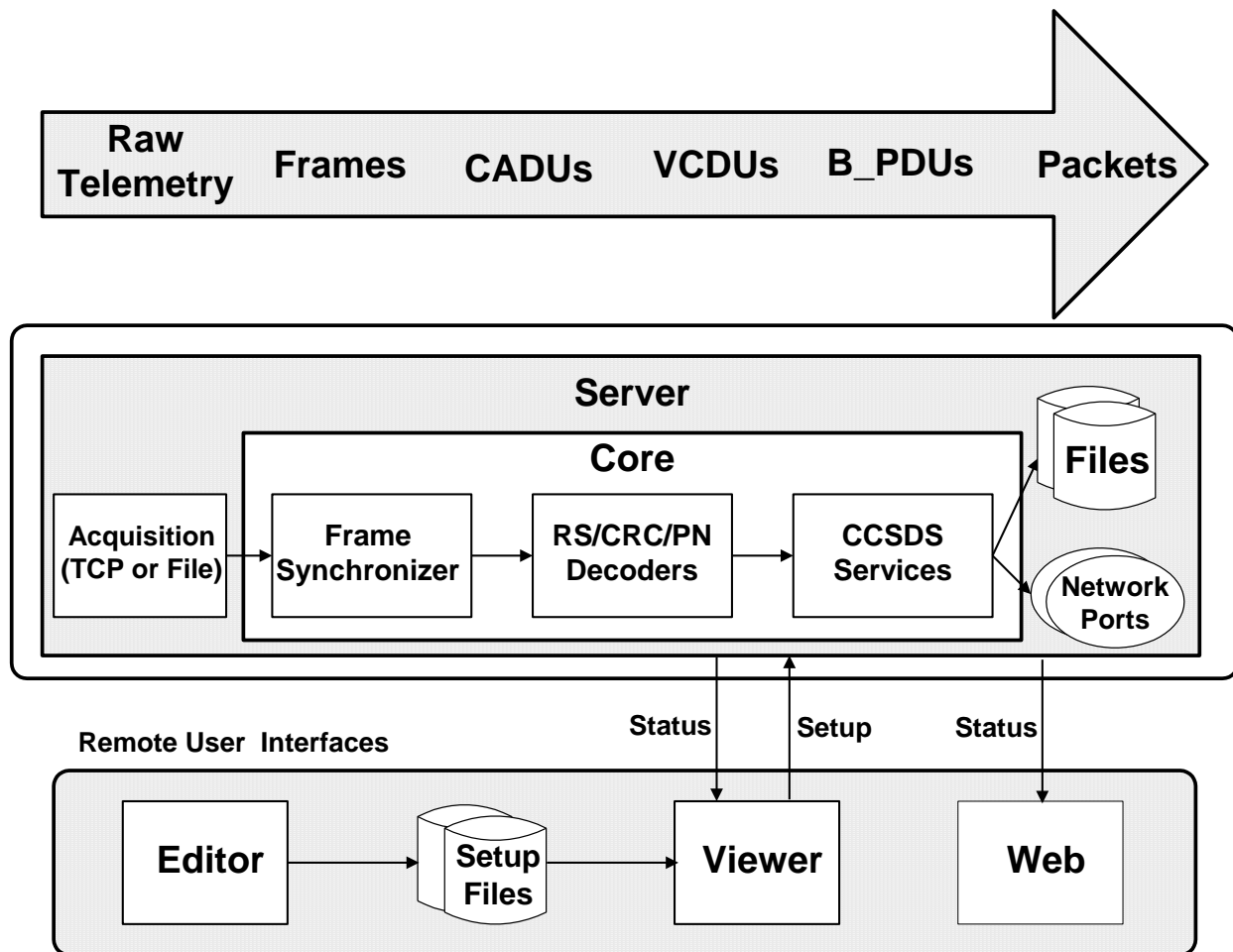


Figure 1. RT-STPS Architecture

3 Software Version

This software package contains RT-STPS Version 7.0. Copyright 1999-2007, United States Government as represented by the Administrator for the National Aeronautics and Space Administration. All Rights Reserved.

Enhancements to Version 7.0 include:

- Added Joint Polar Satellite System 2 (JPSS-2) processing capability for operational and research data models (NOAA – Raw Data Record [RDR] and NASA – Production Data Set [PDS] formats). JPSS-2 test data is available in the RT-STPS_7.0_PATCH_1_testdata.tar.gz file. This test data provides both the raw CADU input to RT-STPS and respective output RDR/PDS. Detailed definition file(s) exist (readme and log file) for easy interpretation of expected results.
- Bundled custom Java Run-time Environment (JRE) for Linux (Oracle OpenJDK 11.0.1) now included within RT-STPS. This architectural change intends to

promote greater portability and reliability given the significant changes in support of JPSS-2.

- Updated HDF5 attributes for RDRs according to latest Common Data Format Control Book (CDFCB) documents for JPSS-2.
- RDR definitions XML files now contain spacecraft-specific attribute definitions.
- Improved logging in support of JPSS-2.
- Augmented Reed-Solomon interleave 5 capability (now needed for JPSS-2).

4 Program Inputs and Outputs

RT-STPS ingests raw CCSDS-compliant frames that may be Pseudo-noise (PN)-encoded or Reed-Solomon (RS)-encoded and outputs VCDUs or packets into the following formats:

- a) RDR files for Suomi National Polar-Orbiting Partnership (SNPP) and Joint Polar Satellite System 1 (JPSS-1) Mission instruments (Visible Infrared Imaging Radiometer Suite [VIIRS], Advanced Technology Microwave Sounder [ATMS], Cross-track Infrared Sounder [CrIS], Ozone Mapping Profiler Suite [OMPS], and Clouds and Earth's Radiant Energy System [CERES]);

RDR files for Joint Polar Satellite System 2 (JPSS-2) Mission instruments [Visible Infrared Imaging Radiometer Suite (VIIRS), Advanced Technology Microwave Sounder (ATMS), Cross-track Infrared Sounder (CrIS), Ozone Mapping Profiler Suite (OMPS)];
- b) Production Data Set (PDS) (packet file and Construction Record [CSR]) file sets for:
 - Earth Observing System (EOS) Aqua Mission instruments (Atmospheric Infrared Sounder [AIRS], Advanced Microwave Scanning Radiometer [AMSR], Advanced Microwave Sounding Unit [AMSU], CERES, and Moderate Resolution Imaging Spectroradiometer [MODIS]);
 - EOS Terra Mission MODIS instrument;
 - SNPP Mission instruments, as listed in item (a) above;
 - JPSS-1 Mission instruments, as listed in item (a) above;
 - JPSS-2 Mission instruments, as listed in item (a) above;
- c) File: header, trailer, and no annotation;
- d) Sockets.

5 Installation and Configuration

RT-STPS can be installed on Linux platforms. Installation and configuration instructions are provided in section 5.1. Program operation instructions for installations are contained

in section 6.

The RT-STPS_7.0_testdata.tar.gz file contains a sample JPSS-1 raw input data file (“rt-stps_jpss1_testdata.dat”) that may be processed with RT-STPS to test RT-STPS following installation and configuration. Instructions are contained in section 5.1.6. The “rt-stps_jpss1_testdata.dat” file will produce RDRs for the JPSS-1 VIIRS, ATMS, CrIS, and OMPS instruments, and PDS file sets for the JPSS-1 Spacecraft, VIIRS, and OMPS instruments. Refer to section 7 for additional information.

5.1 Installation

When installing RT-STPS on a computer where it has been installed previously, be sure to copy any customized configuration files from the previous installation to a safe location so they will not accidentally be overwritten during installation.

If you are upgrading from a previous version of RT-STPS, first stop the RT-STPS services, then delete the existing ‘rt-stps’ directory.

5.1.1 Requirements

The RT-STPS package comes pre-compiled using Oracle OpenJDK 11.0.1 for 64-bit Linux. RT-STPS is bundled and executed with a standalone Oracle OpenJDK 11.0.1 JRE. A 64-bit version of Linux is required for proper operation.

RT-STPS has been tested on a 64-bit computer with a CentOS Linux 7 x86_64 operating system.

5.1.2 Installation Instructions

NOTE: These installation instructions assume that the RT-STPS package is installed in ‘/home/username’.

WARNING: Due to potential resource concerns, RT-STPS should not be installed on a system that is also running an International Polar Orbiter Processing Package (IPOPP) installation.

If the RT-STPS is currently running, stop it by executing the following from the command line:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./jsw/bin/rt-stps-server.sh stop
```

If installing RT-STPS on a computer where RT-STPS has been installed previously, first save any customized configuration files to another location before removing the existing 'rt-stps' directory.

Then to install the new package:

- 1) create a user account (if it does not already exist) under which the server will run;
- 2) copy the downloaded RT-STPS_7.0.tar.gz file to the desired installation directory;
- 3) decompress RT-STPS_7.0.tar.gz using the command:

```
$ tar xzvf RT-STPS_7.0.tar.gz
```

An 'rt-stps' directory containing the contents of the RT-STPS package should now be installed in the selected installation directory. The RT-STPS user must own and have full read/write permissions to the installation and 'rt-stps' directories. Copy any saved configuration files to the new package location as is appropriate for your system.

Copy the downloaded RT-STPS_7.0_testdata.tar.gz file to the installation directory (same directory as the 'rt-stps' directory). Decompress and un-archive the RT-STPS_7.0_testdata.tar.gz file:

```
$ tar -xzf RT-STPS_7.0_testdata.tar.gz
```

5.1.3 Configure RT-STPS

Change to the 'rt-stps' directory and run the installation script:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./install.sh
```

In order to make the script executable, it may be necessary to run the command:

```
chmod +x install.sh
```

The following message will appear:

```
Configuring RT-STPS...
```

When configuration is complete, the user will receive the following message:

```
Configuration complete.
```

This message indicates that an RT-STPS package has been installed and configured successfully.

5.1.4 leapsec File Updates

NOTE: Please ensure that all active RT-STPS server instances are stopped before updating the RT-STPS leapsec file. Refer to Section 6, Program Operation, for details on stopping and starting server instances.

RT-STPS requires a leapsec file to calculate the current number of leap seconds to use in time calculations. An up-to-date leapsec file is required for RT-STPS operation. A leapsec file is included in the RT-STPS software package. For correct time calculations, please ensure that an up-to-date leapsec.dat file is located in the root directory of RT-STPS. Leapsec files are available at:

<https://is.sci.gsfc.nasa.gov/ancillary/temporal/>

The leapsec filename must follow one of the two following formats:

- a) leapsec.dat
- b) leapsec.YYYYMMDDNN.dat, e.g. leapsec.2013042201.dat

The leapsec file can be updated by running the update_leapsec.sh script, as follows:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/internal/update_leapsec.sh
```

5.1.5 Package Layout

Once the package has been unpacked and installed, the 'rt-stps' directory should contain the following contents:

- bin/ – command line applications
- config/ – XML configuration files
- data/ – definition files used for RDR processing
- docs/ – javadoc target directory
- images/ – various icon and graphic images
- info/ – directory where auto-generated server info files are stored
- jre/ - contains the Java Runtime Environment used to execute RT-STPS
- jsw/ – Java Service Wrapper software for Linux
- lib/ – JAR files including rt-stps.jar and HDF libraries
- logs/ - contains log files generated by the “server.sh” script
- testdata/ – contains the “rt-stps_jpss1_testdata.dat” file and test output products
- util/ – contains RT-STPS utilities
- install.sh – installation script for Linux
- leapsec.dat – leapsec file used for accurate time calculations
- rt-stps.dtd – the Data Type Definition (DTD) for the XML configuration files
- rt-stps.policy – policy statement
- SUA Open Source IPOPP GSC-15570-1.pdf – software use agreement
- VERSIONLOG – version info

5.1.6 Testing the Installation

The server sets the 'rt-stps' directory as the root directory (RT-STPS files used by the server are searched starting from the root directory), and the configuration files specify that output will go to the './data' directory. RT-STPS places output into the 'data' directory at the same level as the 'rt-stps' directory (i.e., './data').

After testing the installation the user may wish to edit the configuration files to specify a preferred destination directory.

5.1.6.1 Start the Server Manually

To test the installation the server must be started from the command line. Ensure that you are in the 'rt-stps' directory and then start the server by issuing the start command to the JSW (Java Service Wrapper):

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./jsw/bin/rt-stps-server.sh start
```

The JSW captures all output messages to a log file in 'rt-stps/jsw/logs'. The overall status of the server can also be checked by issuing the status command to the JSW:

```
/home/username/rt-stps> ./jsw/bin/rt-stps-server.sh status
```

This command will return its current process identifier and a message confirming it is running.

5.1.6.2 Execute the Viewer

Next start the viewer application. The viewer allows the user to load configuration files and to view processing status. From the command line in the 'rt-stps' directory enter:

```
/home/username/rt-stps> ./bin/viewer.sh &
```

This command will bring up the viewer Graphical User Interface (GUI), depicted in Figure 2. Use the Load button to load the "jpss1.xml" configuration file from the 'rt-stps/config' directory. Use the file dialog to find this directory and file. Once the file loads, click on the Go button.

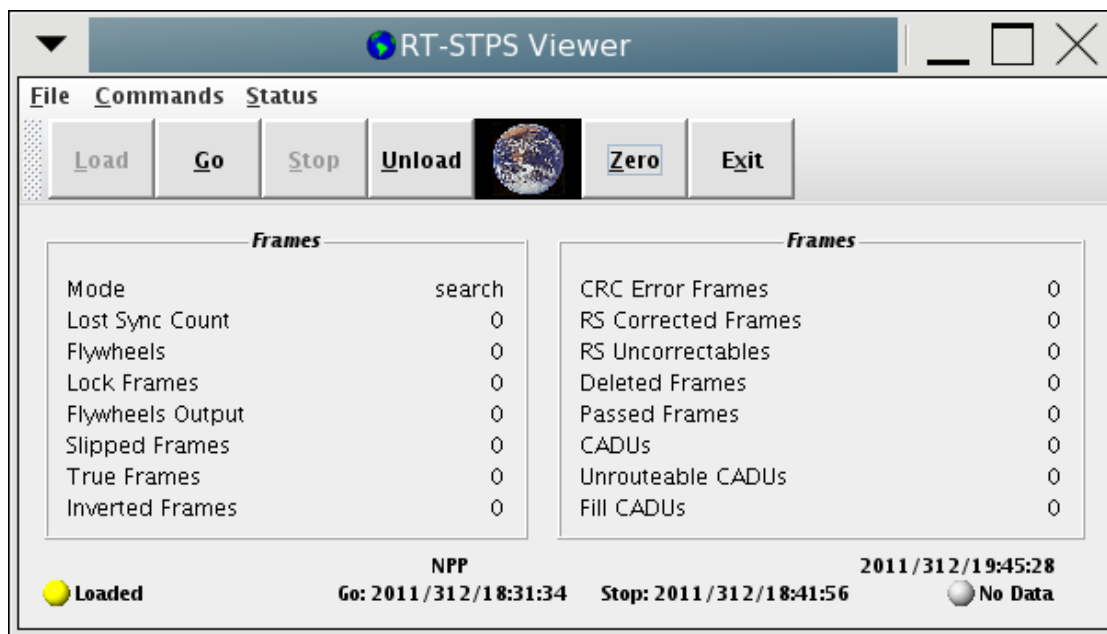


Figure 2. RT-STPS Viewer

5.1.6.3 Execute the Sender

Next start the sender application. The sender allows the user to send telemetry data files to the RT-STPS server for processing. It also displays the percentage of the file that has been sent to the server. From the command line in the 'rt-stps' directory, enter:

```
/home/username/rt-stps>./bin/sender.sh &
```

This will bring up the sender GUI.

Click on the File button. Use the File dialog to select the "rt-stps_jpss1_testdata.dat" file for JPSS-1, from the 'testdata/input' directory extracted from the RT-STPS_7.0_testdata.tar.gz file, and click on the Go button to send the file to the server for processing. The sender is depicted in Figure 3.

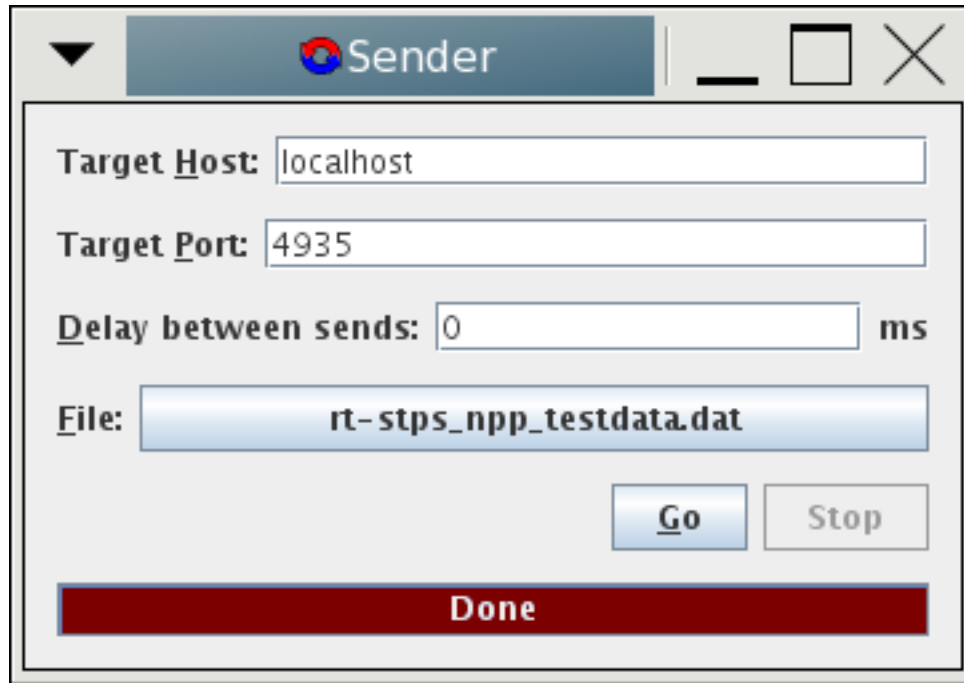


Figure 3. RT-STPS Sender

5.1.6.4 Inspect the Results

Once processing is complete (the viewer status buttons will show that no more data are being processed, and the sender will show that it is done), list and inspect the contents of the './data' directory. It should contain RDR and PDS files with filenames similar to the following (the dates of the creation fields will be different):

```
P1590000AAAAAAAAAAAAAS17282170443000.PDS
P1590000AAAAAAAAAAAAAS17282170443001.PDS
P1590008AAAAAAAAAAAAAS17282170443000.PDS
P1590008AAAAAAAAAAAAAS17282170443001.PDS
P1590011AAAAAAAAAAAAAS17282170443000.PDS
P1590011AAAAAAAAAAAAAS17282170443001.PDS
P1590608AAAAAAAAAAAAAS22067172832000.PDS
P1590608AAAAAAAAAAAAAS22067172832001.PDS
P1590609AAAAAAAAAAAAAS22067172832000.PDS
P1590609AAAAAAAAAAAAAS22067172832001.PDS
P1590826VIIRSSCIENCEAS17282170443000.PDS
P1590826VIIRSSCIENCEAS17282170443001.PDS
RATMS-RNSCA_j01_d20160722_t1114034_e1130353_b00001_c20171009170801799000_all-_dev.h5
```

RCRIS-RNSCA_j01_d20160722_t1114034_e1130353_b00001_c20171009170802119000_all-_dev.h5
RNSCA-RONPS_j01_d20160722_t1114034_e1130533_b00001_c20171009170801779000_all-_dev.h5
RNSCA-ROTCS_j01_d20160722_t1114034_e1130533_b00001_c20171009170801437000_all-_dev.h5
RNSCA-RVIRS_j01_d20160722_t1114001_e1131043_b00001_c20171009170801812000_all-_dev.h5

Test output products are available in the 'rt-stps/testdata/output' directory. The test output products serve as an indicator of expected program output. Use a comparison utility (such as diff, h5diff, etc.) to compare your output products to those provided in the 'rt-stps/testdata/output' directory. Locally generated products may differ slightly from the provided test output products because of differences in file creation times, machine architecture, or operating systems.

5.1.6.5 Stop the Server

Once the test is complete the server may be stopped by issuing the stop command to the JSW:

```
/home/username/rt-stps>./jsw/bin/rt-stps-server.sh stop
```

WARNING: Since the viewer/sender requires the RT-STPS server to be running, it is advised that the sender and viewer be closed prior to stopping the server.

5.1.7 Creating Launchers

Desktop launchers to run the viewer and sender may be created to start these applications. The creation of launchers varies between the different types of Linux distributions and their desktop environments. In all cases the commands described in section 5.1.6.2 and section 5.1.6.3 will apply.

5.1.8 Firewall Configuration

A firewall may in some cases prevent RT-STPS from running correctly, and it may need to be disabled or configured to allow access for certain ports. The server accepts data by default on port 4935. The output port numbers, if any, are defined in the configuration files. The viewer initially connects through port 1099; afterwards, the viewer and server communicate through anonymous ports. The server also uses port 5935 for the alternate server interface.

5.2 Java Setup and Performance Notes

RT-STPS now comes bundled with, and runs on, a custom Java runtime environment. This runtime environment exists in the 'rt-stps/jre' directory. Many of the bash scripts in the RT-STPS 'bin' directory employ the "-server" option to increase performance, as the Java Virtual Machine (JVM) server version may be somewhat faster than the client version of the JVM. If the "-server" option is used in a script file and the server JVM is not properly installed, the script file will not run properly and terminate with an error.

6 Program Operation

RT-STPS can be started as a server from a command line server script, or from the JSW.

A configuration file is required to describe the telemetry format specifications and the data outputs for the spacecraft of interest. Typically, the setup configuration file will be loaded prior to each pass. Sample setup files are stored in the 'rt-stps/config' directory.

The './data' directory is the default target for output file results for RT-STPS.

A sample raw telemetry file named "rt-stps_jpss1_testdata.dat" resides in the 'testdata/input' directory extracted from the RT-STPS_7.0_testdata.tar.gz file. It contains packet data from the VIIRS, ATMS, CrIS, and OMPS instruments aboard the JPSS-1 satellite. A corresponding "jpss1.xml" configuration file resides in the 'config' directory. Use it to produce NASA-model Spacecraft, VIIRS, and OMPS PDS files, and VIIRS, ATMS, CrIS, and OMPS science RDRs. Instructions for testing RT-STPS using these sample files are contained in sections 5.1.6. The RT-STPS package also includes configuration files corresponding to other spacecraft. Refer to section 6.1 for more details.

6.1 Provided Configuration Files

The 'rt-stps/config' directory contains pre-made RT-STPS XML configuration files corresponding to all currently-supported spacecraft. The sample XML configuration files are as follows:

- a) default.xml – default configuration loaded by RT-STPS if a configuration file is not otherwise specified.
- b) aqua.xml – reads Aqua spacecraft telemetry and writes PDS (packet file and Construction Record [CSR] file pairs) data to files and a port for each of the major instruments aboard the spacecraft.
- c) terra.xml – reads Terra spacecraft telemetry and writes PDS data to files and a port for the MODIS instrument.
- d) npp.xml – reads Suomi NPP spacecraft telemetry and will produce Science RDRs for the ATMS, CrIS, OMPS, VIIRS, and CERES instruments, with or without attitude and ephemeris (depending upon their presence in the data). NASA-model Spacecraft, VIIRS, and OMPS PDS files are also generated.
- e) jpss1.xml – reads JPSS-1 spacecraft telemetry and will produce Science RDRs for the ATMS, CrIS, OMPS, VIIRS, and CERES instruments, with or without attitude and ephemeris (depending upon their presence in the data). NASA-model Spacecraft, VIIRS, and OMPS PDS files are also generated.
- f) jpss2.xml – reads JPSS-2 spacecraft telemetry and will produce Science RDRs for ATMS, CrIS, OMPS, and VIIRS instruments, with or without attitude and ephemeris (depending upon their presence in the data). NASA-model Spacecraft, VIIRS, and OMPS PDS files are also generated.

6.2 Starting the Server

The server may be started from a command line server script, or from the JSW. It is strongly recommended to start the server from the JSW.

6.2.1 Command Line Script

Change to the 'rt-stps' directory and run the server script:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/server.sh &
```

6.2.2 Java Service Wrapper

Change to the 'rt-stps' directory and issue the start command to the JSW:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./jsw/bin/rt-stps-server.sh start
```

These commands initiate the server process in the background. When ready, the "Ready to serve" status message is logged to the appropriate location. Refer to section 6.6 for details regarding logging.

6.3 Viewer

The viewer may be run from the command line or as a shortcut from the desktop if this has been configured on your system. To run from the command line, change to the 'rt-stps' directory and execute the viewer script:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/viewer.sh &
```

The viewer GUI will appear on the screen (see Figure 2). Use the viewer to configure the server and examine its status, as well as to load and unload configuration files. Viewer functions are summarized below.

Menu Bar

The Menu Bar contains the File, Commands, and Status pull-down menus.

File Menu

The File Menu contains the program Exit item.

Commands Menu

The Commands Menu contains the following commands to configure and run the server:

- a) Load. Displays a Dialog Box to select and load a configuration file stored on the computer executing the viewer.
- b) Go. Starts server data processing.

- c) Stop. Halts server data processing.
- d) Unload. Removes the current configuration file from the server.
- e) Zero Status. Resets the statistics display.
- f) Exit. Exits the Viewer.

Status Menu

The Status Menu contains menu items to display Path Service Status, Packet Status, the Virtual Channel Status Table, and the Packet Status Table.

Button Bar

The button bar contains buttons linked to the Commands Menu items (see Figure 2).

6.4 Sender

The RT-STPS sender (Figure 3) is used to send a raw data file to the server. The sender may be run from the command line or as a shortcut from the desktop if this has been configured on your system. To run from the command line, changed into the 'rt-stps' directory and execute the sender script:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/sender.sh &
```

The Target Host should be "localhost", the Target Port number should be "4935", and the delay between sends should be zero. Click on the File button. Use the File dialogue to select the desired raw data file, and click on the Go button to send the file to the server for processing. The server status may be checked using the RT-STPS viewer. When the sender finishes, the server will automatically halt data processing and unload the current configuration file.

6.5 Batch Mode

A batch command performs server processing as a standalone, one-time program. It takes as arguments a configuration file and an input data file containing telemetry frames. The outputs are specified in the configuration file. At the command line, change to the 'rt-stps' directory and run the batch script:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/batch.sh <path-to-config-file> <path-to-data-file>
```

6.6 Logging

There are two types of RT-STPS output messages: status/event messages, and standard output. RT-STPS logs these messages depending on how the RT-STPS server was started.

Server status/event messages are logged into the “rt-stps-server.log” file in the ‘rt-stps/jsw/logs’ (for the JSW server) or ‘rt-stps/logs’ (for command line server) directory by default.

Standard output messages generated by the RT-STPS server are only logged if the server is started from the JSW. Standard output messages are logged in rolling log files named “rt-stps-server.log.*” in the ‘rt-stps/jsw/logs’ directory, which are automatically managed by the JSW.

6.7 Stopping the Server

The server may be stopped from a command line script or from the JSW, depending on how it was started. Because the sender and viewer require the RT-STPS server to be running, it is advised that the sender and viewer be stopped prior to stopping the server.

6.7.1 Command Line Script

If the server was started using the command line server script, change to the ‘rt-stps’ directory and run the stop script:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/stop.sh
```

6.7.2 Java Service Wrapper

If the server was started from the JSW, change to the ‘rt-stps’ directory and issue the stop command to the JSW:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./jsw/bin/rt-stps-server.sh stop
```

6.8 Automatic Setup

If the server is running but no configuration file is loaded and data arrives on its incoming data port (port 4935), then the server will use the “default.xml” file as its configuration file.

Similarly, if the server is running but has not been reconfigured with a new configuration file, and data unexpectedly arrive on its incoming data port, then the server will use the previous configuration file.

If the server is shutdown, it will revert to using the “default.xml” file when next started.

7 Raw Data Record (RDR) Creation

7.1 Supported RDRs

RT-STPS Version 7.0 supports the creation of Raw Data Records (RDRs) for the Suomi NPP and JPSS-1 VIIRS, ATMS, CrIS, OMPS, and CERES instruments. It also supports the creation of RDRs for the JPSS-2 VIIRS, ATMS, CrIS, and OMPS instruments. Corresponding Spacecraft Diary data is produced from attitude and ephemeris packets included in the input, and included in the produced RDRs.

7.2 Test Data

The RT-STPS_7.0_testdata.tar.gz file contains a sample JPSS-1 raw input data file named “rt-stps_jpss1_testdata.dat”.

These files may be processed with RT-STPS to produce RDRs for the JPSS-1 VIIRS, ATMS, CrIS, and OMPS instruments.

7.3 Suomi NPP, JPSS-1, and JPSS-2 Configuration File for RDRs

RT-STPS includes the “npp.xml”, “jpss1.xml”, and “jpss2.xml” configuration files that define everything necessary to create the Suomi NPP and JPSS-1 VIIRS, ATMS, CrIS, OMPS, and CERES (CERES not on JPSS-2) RDRs respectively, as well as to output VIIRS science packets to a socket interface. The “npp.xml”, “jpss1.xml”, and “jpss2.xml” configuration files can be used as the basis to create other RDR configurations of interest (e.g., a configuration file dedicated solely to VIIRS Science RDRs, among others). The RDR creation process is depicted in Figure 4.

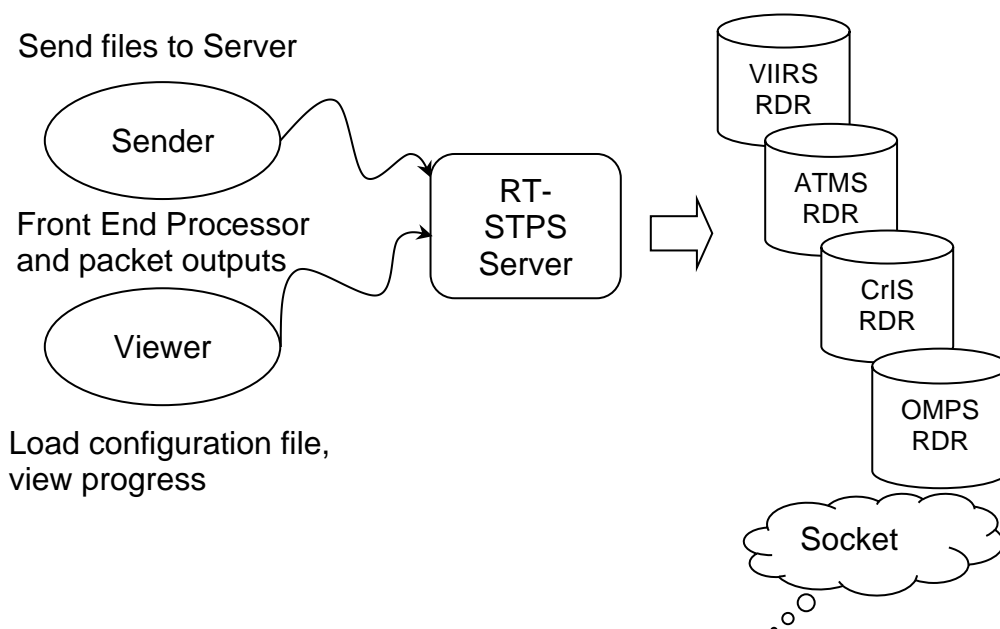


Figure 4. RDR Creation Process

7.4 Processing the Test Data

Decompress and un-archive the RT-STPS_7.0_testdata.tar.gz file. Refer to section 5.1.6.1.

Use the viewer to load the configuration file. Refer to section 5.1.6.2 for instructions to execute the viewer. Use the sender to send the 'rt-stps_jpss1_testdata.dat' file to the server for processing. Refer to section 5.1.6.3 for instructions to execute the sender.

The server uses the './data' directory by default to store processed results. Refer to section 5.1.6.4 for additional details.

7.5 RDR Processing Status

When RT-STPS is run in server mode, the RDR processing module updates some status information as it creates granules for the RDR of interest. The status information may be viewed using the RT-STPS viewer, or by using the getstatus script (refer to section 8.1). In batch mode, this information is output to the console directly. Refer to section 6.5 for instructions to run RT-STPS in batch mode.

7.6 Expected Output

Once processing is complete, the viewer status buttons will show that no more data is being processed, and the sender will show that it is done. The './data/' directory should contain the generated RDR files. Refer to section 5.1.6.4 for more details regarding the expected output when processing the "rt-stps_jpss1_testdata.dat" raw data file.

7.7 Reference Sources

Details on mission data specifications are contained in the following Joint Polar Satellite System (JPSS) documents:

- Joint Polar Satellite System (JPSS) Common Data Format Control Book - External Volume I – Overview
- Joint Polar Satellite System (JPSS) Common Data Format Control Book - External Volume II – RDR Formats;
- Joint Polar Satellite System (JPSS) Common Data Format Control Book - External Volume V – Metadata;
- Joint Polar Satellite System (JPSS) Algorithm Specification Volume II: Data Dictionaries– Block 2.0.0 – Parts 1-9, 28

Information to identify, distinguish and extract all of the X-band unique source packets from the Suomi NPP, JPSS-1, and JPSS-2 mission data streams is contained in the respective Joint Polar Satellite System (JPSS) Mission Data Format Control Book (MDFCB).

These documents are available via the DRL Web Portal at:
<https://directreadout.sci.gsfc.nasa.gov/?id=dspContent&cid=16>

8 Additional Command Line Tools

Additional scripting tools in the 'rt-stps/bin' directory provide additional functionality to RT-STPS.

8.1 Getstatus

The getstatus script periodically retrieves status information from the active RT-STPS server on the local host until it is stopped. The format is:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/getstatus.sh
```

8.2 Load

The load script loads a local configuration file into the local server. The format is:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/load.sh <path-to-config-file>
```

8.3 Shutdown

The shutdown script halts data processing and unloads the current configuration file on the local server. The format is:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/shutdown.sh
```

When the script completes, the RT-STPS Sender (if used) will display a "Connection reset" pop-up window.

8.4 Version

The version script prints RT-STPS version information and exits. The format is:

```
/home> cd /home/username/rt-stps  
/home/username/rt-stps> ./bin/version.sh
```

8.5 Rate Buffering Program

RT-STPS includes a rate-buffering program (Rat) that can spool data to a slow target (e.g., such as through a slow network). The server may run slowly because all outputs are processed in one control loop, without internal buffering or internal multi-threading. By using the Rat as an intermediate destination, the server can keep one slow target from slowing the outputs of other targets being serviced at the same time.

Rat may be run on the same computer as the server, or from a remote location. It requires three arguments when it is invoked:

```
/home> cd /home/username/rt-stps
```

```
/home/username/rt-stps>./bin/rat.sh <inputPort> <targetHost> <targetPort> &
```

Rat listens for socket connections on its input port. When a connection is made, it connects to the target on the target port, forming an end-to-end connection. However, it can only service one input connection at a time. Once connections are established, Rat will buffer input data while sending some data to the slower output connection. Once the input connection closes, Rat will continue to send buffered data to the target until it has no more data, at which point it will close the target connection. Rat forms end-to-end connections after the RT-STPS server has connected to it and will run continuously, listening for and accepting connections until it is terminated.

To stop the Rat program, find its process ID and use the appropriate tools/commands to terminate it. For example:

Search for the correct rat.sh process ID, and terminate it. Use commands such as “ps aux” and “grep” to find the correct “gov.nasa.gsfc.drl.rtstps.clients.spooler.Main” process (the one with the desired input port, target host, and target port). For example:

```
/home/username/rt-stps> ps aux | grep "gov.nasa.gsfc.drl.rtstps.clients.spooler.Main  
<inputPort> <targetHost> <targetPort>"  
/home/username/rt-stps> kill <Process-ID>
```

(where <inputPort>, <targetHost>, and <targetPort> are the target host and port specified when the rat process was started)

9 Alternate Server Interface

There is an additional command interface to the server accessible by software. This interface is useful to integrate the RT-STPS into a larger system with customization (e.g., connect it to scheduling software that may be modified to run RT-STPS).

The interface is through port 5935 by default. The server expects to receive certain text string messages on this port for control. Each message should have the normal line terminator, and case is significant. The messages are commands to load and shut down sessions. The available commands are:

- a) loadgo <configurationFileName> - the server will load the configuration file from its configuration directory, and then enables itself for processing (e.g., “loadgo npp.xml”);
- b) shutdown - stops processing and unloads the current configuration, which closes all output files.

The server does not send responses to any of these commands. The only feedback is via the server’s console window. It will print the usual load and shutdown messages. It will also print error messages labeled as “ProxyThread” messages if it encounters them.