

STEREO Science Center Operations Manual

William Thompson

September 13, 2022

Contents

1	Introduction	2
2	Beacon telemetry	3
2.1	Tracking antenna performance	4
2.2	In-situ and radio beacon data	5
2.3	SECCHI beacon data	5
2.4	SECCHI beacon browse products	6
3	Browse pages	7
4	MOC data products	8
5	In situ data and software	9
5.1	IMPACT data and software	10
5.2	PLASTIC data and software	10
5.3	Status pages	10
5.4	S/WAVES data	10
5.4.1	S/WAVES Level 2 data	11
6	SECCHI data	11
6.1	HI Level 2 data	12
6.2	EUVI wavelet images	12
7	COR1	13
8	SPICE ephemerides	14
9	STEREO Orbit Tool	14
10	Miscellaneous scripts	15
10.1	Scripts on moat	15

1 Introduction

This document describes the structure of the STEREO Science Center (SSC), and the various scripts which run under the account “wtthomps” (or “thompson” depending on the computer), and which are used to keep the data and web pages up-to-date. Some other topics which are related to, but not directly part of, SSC operations are also covered.

The SSC consists of four workstations that are connected to the Solar Data Analysis Center (SDAC), along with the STEREO-specific filesystems within the SDAC archive. These four workstations are:

beacon: A virtual workstation used for processing the STEREO beacon data. This virtual workstation runs an older version of Mac OSX to accommodate software used to process the SECCHI beacon images which was originally compiled for the PowerPC platform. Plans are underway to migrate this machine to Linux, at which point it will no longer be a virtual machine.

anaglyph: Used for all processing other than the realtime beacon data.

stereodata: Web interface to the STEREO data. Mounts the archive in read-only mode. Web pages can be edited from anaglyph or beacon in the directory `/service/stereo3/stereo/www/htdocs`.

ssc: Performs backups.

All of the above computers are on the SDAC subnet of the NASCOM network. There are also two computers on the GSFC (NDC) network:

hifi: Secondary copy of the primary part of the archive. This copy is complete for the in situ and radio data, but only contains the Level-0.5 FITS files for SECCHI. Also serves these files via NFS to computers within the GSFC domain in Building 21. Plans are underway to migrate this to new hardware with larger storage.

stereo: Serves additional web pages, with emphasis on public outreach. Often referred to as “stereo.gsfc”. The web pages are mounted read-only, and are maintained on the workstation hifi in the directory `/service/stereo1/stereo`.

For historical reasons, the scripts on the SSC computers which address the computers on the GSFC domain use the term “bldg21” because the other computers were originally in building 26, back when the SDAC was located there.

Except for the telemetry ingestion routines discussed in Section 2, all of the software discussed in this document can be found in `/service/stereo3/stereo/software`. For convenience in the discussion below, we will describe this as `$SOFTWARE`, even though this is not an actual environment variable used in the software. Most scripts are in the subdirectory `$SOFTWARE/scripts`. Each script generates a `.log` file in the same directory as the script.

Most of the IDL routines used by the various scripts are found in the Solar Software (SSW, or SolarSoft) library. Routines specific to SSC are in the `$$$SW/stereo/ssc` branch of SolarSoft. The parallel Solar Software Database (`$$$WDB`) is used to contain large data files.

A small number of scripts that require access to non-STEREO parts of the SSW tree run on the workstation `moat.nascom.nasa.gov`, which has access to the entire SSW tree.

2 Beacon telemetry

The STEREO beacon is a low 640 bps telemetry stream broadcast by the spacecraft 24 hours a day. During DSN contacts, the beacon telemetry packets are embedded within the overall real-time telemetry stream. These packets are collected from the STEREO Mission Operations Center (MOC) through a socket connection to the MOC telemetry server.

Outside of DSN contacts, a network of volunteer antenna stations around the world collect the telemetry and send it in realtime to the SSC via a socket interface. The workstation “beacon” acts as the server, and the various antenna stations connect to this server when they are receiving telemetry from one of the STEREO spacecraft. These data are transferred as telemetry frames rather than packets.

A separate interface was used in the past to collect telemetry from the French station in Toulouse. The realtime telemetry frames were placed on an FTP server and pulled by the SSC. The Toulouse station no longer supports STEREO, so this interface is no longer used.

The software used by the volunteer ground stations was developed by NOAA, and (in principle) NOAA has the responsibility of organizing the ground stations. The mail list `stereo.ops@noaa.gov`, managed from NOAA, can be used to communicate with the ground stations. The current point-of-contact with NOAA is Rob Craver (`rob.craver@noaa.gov`), and the email address `rwc.boulder@noaa.gov` is a mailing list for the forecasters.

Unlike the other scripts discussed in this document, including those discussed in the various subsections below, all the scripts used for *collecting* the beacon telemetry run under the root account and are managed by the system administrator Marc Despres. These scripts separate out the telemetry frames into their individual packets, organize the packets by spacecraft and instrument, and merge the packets from the various sources into single streams, time-ordered, and with duplicates removed. The beacon telemetry and software are found in `/service/stereo1/swx_telem`. The subdirectory `software` contains the software routines that collect the telemetry.

The subdirectory `data` contains the collected telemetry, organized by spacecraft (ahead or behind), year, and month. Beacon telemetry are stored in two formats, `.ptp` and `.stp`—the latter contains additional information about the telemetry frames that originally contained the packets. Note that the IDL routines `read_stereo_pkt` and `parse_stereo_pkt` work with both file formats. An example of a beacon telemetry file is

```
.../swx_telem/data/ahead/2021/04/secchi_sta_2021_102_040002_00.sw.ptp
```

This file contains SECCHI beacon packets from the STEREO Ahead (STA) spacecraft, for day-of-year 102 (April 12) of the year 2021, for packet times starting at 04:00:02 UT.

The subdirectory `header` is similar to `data` with two differences. First, the files in this directory tree contain only the packet headers, not the complete packets. Also, duplicate packets are *not* removed in these files. The purpose of the `header` tree is to track the performance of the various antenna stations, based on the information in the packet headers. Thus, duplicate packets resulting from overlapping coverage from two (or more) stations are retained, so that all the coverage from each individual station can be tracked, even if packets from another station arrive first.

Additional scripts, described below, read in the telemetry files and process them into beacon data products. These scripts all run under the account “thompson”.

2.1 Tracking antenna performance

The ground-receipt-header portion of the telemetry packet structure contains a field for the DSN antenna ID. For the beacon telemetry collected by the SSC, this field is used to store ID numbers representing the various antenna partner stations. At the present time, this field contains the following values:

- 1 DSN (regardless of station)
- 14 Chilbolton, England
- 15 Koganei, Japan
- 16 Toulouse, France
- 17 Kiel-Ronne, Germany
- 18 Bochum, Germany
- 19 KSWC, Korea
- 20 APL, Maryland, USA

The webpage `stereo-ssc.nascom.nasa.gov/beacon/beacon_coverage.shtml` contains a plot of the coverage provided by the various antenna stations over the last 24+ hours. This plot is produced by the script `SOFTWARE/beacon/ssc_beacon_coverage`, using the files in the `header` directory tree described above. The associated script `start_ssc_beacon_coverage` is used to start the script at boot time. The files `ssc_beacon_coverage.host` and `ssc_beacon_coverage.pid` contain the hostname and IDL process ID—these files are used by the associated script `restart_ssc_beacon_coverage` to decide whether or not the script needs to be restarted after a crash. Emails are sent automatically whenever the script crashes, and also when it restarts.

Note: If any script in this document with associated .host and .pid files becomes “hung”, or otherwise does not appear to be progressing as it should, the process ID can be used to help clear the problem. In some cases the problem may not be in the main process, but in a subprocess. The command `ps -g (MacOS)` or `ps -ejH | fgrep -i !` (Linux¹) followed by the process ID can be used to list all the subprocesses associated with the main process. Killing a hung subprocess may allow the main process to continue, or it may be necessary to kill both the main process and all its subprocesses.*

The web page `stereo-ssc.nascom.nasa.gov/beacon/swx_daily_coverage.dat` contains the percentage of each day covered by the realtime beacon telemetry for each spacecraft. This file is produced by the script `SOFTWARE/scripts/make_daily`. The page `stereo-ssc.nascom.nasa.gov/beacon/station_report.txt` contains a report of how many

¹Aliased as “psminusg” in the wthomps account on anaglyph.

packets were collected by each station over the previous month, and is produced by the script `make_station_report`.

2.2 In-situ and radio beacon data

A single script, `SOFTWARE/beamon/ssc_beacon_reader`, processes the beacon telemetry from the IMPACT, PLASTIC, and S/WAVES instruments for both spacecraft. The data from the IMPACT and PLASTIC instruments are processed into CDF files in the directory tree `/service/stereo3/stereo/data/beamon`. S/WAVES data are stored as IDL save files in the same directory tree. At the same time, realtime plots of these data are written to `/service/stereo3/stereo/www/htdocs/beamon`, where they are visible on the STEREO webpage at `stereo-ssc.nascom.nasa.gov/beamon/beamon_insitu.shtml`. The script `update_bldg21_beacon` mirrors these files to the secondary “stereo.gsfc” website.

The associated script `start_ssc_beacon_reader` is used to start the in situ and radio beacon processing at boot time. The files `ssc_beacon_reader.host` and `ssc_beacon_reader.pid` contain the hostname and IDL process ID—these files are used by the associated script `restart_ssc_beacon_reader` to decide whether or not the script needs to be restarted after a crash. Emails are sent automatically whenever the script crashes, and also when it restarts. It should be noted that this script typically crashes about once a day with a “Stale NFS file handle” error. This is due to updates to the PLASTIC calibration files, and the process should restart automatically within the next hour. If this script crashes multiple times in the same day, then something else may be the problem.

2.3 SECCHI beacon data

Processing of the SECCHI beacon is a multistep process using a combination of binary programs, Perl scripts, and IDL routines. The first first step in the process extracts the compressed SECCHI images and headers from the telemetry, and prepares them for further IDL processing. All the scripts for this step in the process are found in `SOFTWARE/secchi`. There are two parts to this process. The actual processing is done by a binary executable called `file_capture`. This executable is compiled for the PowerPC platform, and is the reason that the computer “beacon” requires an older version of Mac OSX.² The Perl script `secchi_sw.pl` supports `file_capture` by reading the beacon telemetry files and feeding the packets to `file_capture` via a socket connection.

There are separate scripts for the Ahead and Behind spacecraft. For simplicity, only the Ahead scripts will be defined. These are

`file_capture_ahead`: Runs the `file_capture` program.

`perl_secchi_ahead`: Runs the supporting Perl script.

`start_file_capture_ahead`: Starts both of the above with a slight pause, so that they run together.

If the processing needs to be restarted for any reason, killing the Perl script will also cause the corresponding `file_capture` process to exit. One should normally never kill `file_capture` directly. Emails are sent automatically when either the `file_capture` or Perl script exits. There is

²New executables are available, and “beacon” will be migrated to new Linux hardware in the near future.

no automatic restart script for this part of the processing. Instead, the software would need to be restarted by hand with the `start_file_capture_ahead` script, first making sure that both of the previous processes are no longer running.

The data extracted from the telemetry packets are written to the directory tree `$$SOFTWARE/file_capture/a` (or `/b`). The next step in the process is to process these data into FITS files. This is done by the IDL routine `secchi_reduce.pro`. The script for this step is `secchi_idl_ahead`, and is found in the same `$$SOFTWARE/secchi` directory as the above scripts. Another important file is `spwximga.env` (or `spwximgb.env` for Behind) which contains the environment variables used by the IDL routines. The FITS files are written to same directory tree as the in situ data (see Section 2.2). The associated script `start_secchi_idl_ahead` is used to start the script at boot time. The files `secchi_idl_ahead.host` and `secchi_idl_ahead.pid` contain the hostname and IDL process ID—these files are used by the associated script `restart_secchi_idl_ahead` to decide whether or not the script needs to be restarted after a crash. Emails are sent automatically whenever the script crashes, and also when it restarts.

As each SECCHI image is processed, the raw data is moved from `$$SOFTWARE/file_capture/a` into the `attic` subdirectory. Occasionally, the IDL processing runs into an image which it cannot process, and repeatedly crashes and restarts. When this happens, the file `secchi_idl_ahead.log` file is examined to determine which dataset is causing the problem. Each dataset consists of multiple files. For example, if the problematic dataset is “E80623GQ”, it may include the following files:

```
E80623GQ-725.dat
E80623GQ-725.hdr
E80623GQ-725.img
E80623GQ-725.tlr
E80623GQ.725
E80623GQ.725.meta
```

All of these files would then need to be moved to the `attic`. Also, the NRL team should be informed so that they can debug the issue and update the software to avoid such issues in the future.

The processing of the SECCHI beacon telemetry is run from a frozen copy of SolarSoft in the directory `$$SOFTWARE/solarsoft`. Typically, this directory tree is maintained by replacing individual routines by hand. However, on *rare* occasions, the script `update_secchi_from_ssw` can be used to completely refresh this directory tree based on what’s currently in SolarSoft. When this script is invoked, it first makes a gzipped tar backup in the directory `$$SOFTWARE/solarsoft_tar`.

There are a number of executables that `secchi_reduce.pro` depends on for processing the files created by `file_capture`. These routines are found in the directory tree `$$SOFTWARE/secchi/utils`, though it is the logical links in `$$SOFTWARE/secchi/bin` which control which executables are called. (The rest of the files in the `bin` subdirectory can be ignored.) Not all of the software in the `utils` tree are used for beacon processing.

2.4 SECCHI beacon browse products

A separate IDL process takes the SECCHI beacon FITS files from both spacecraft and converts them into JPEG browse images. This is done by the script `$$SOFTWARE/beacon/secchi_beacon_images`. The associated script `start_secchi_beacon_images` is used to start the script at boot time. The

files `secchi_beacon_images.host` and `secchi_beacon_images.pid` contain the hostname and IDL process ID—these files are used by the associated script `restart_secchi_beacon_images` to decide whether or not the script needs to be restarted after a crash. Emails are sent automatically whenever the script crashes, and also when it restarts.

Each SECCHI FITS image is converted into JPEG images at various resolutions, depending on the telescope. JPEG files with image sizes of 128×128 , 256×256 , 512×512 , and 1024×1024 are generated for all telescopes, regardless of the size of the original image. For EUVI and COR2, 2048×2048 versions are also produced.

The beacon JPEGs are written into the same directory tree structure used for the STEREO browse pages (see Section 3). The top of the browse directory tree is found at `/service/sterео5/browse`, and appears on the web as `stereo-ssc.nascom.nasa.gov/browse/`. The browse page products are organized by year, month, day, spacecraft, telescope product, and resolution, e.g.

```
.../browse/2021/04/27/ahead/cor2/1024/20210427_162435_n7c2A.jpg
```

points to a 1024×1024 COR2 image. More information about the STEREO browse pages can be found in Section 3.

The web page `stereo-ssc.nascom.nasa.gov/beacon/beacon_secchi.shtml` points to the latest images for each telescope. For some telescopes or EUVI wavelengths which don't come down in the beacon telemetry, the latest images are from previous days. The subdirectories `.../beacon/latest_*` contain logical links to the most recent images at the different resolutions.

Along with the individual JPEG images, the SECCHI beacon processing also generates time-elongation plots, also known as J-plots, based on the most recent data from the COR2, HI1, and HI2 telescopes. These files appear on the website as `stereo-ssc.nascom.nasa.gov/beacon/secchi_jplot.shtml`, and are produced in conjunction with the JPEG images.

Additional beacon products are produced asynchronously by the script `SOFTWARE/scripts/make_browse` running as a cron job on the workstation “anaglyph” every 15 minutes. These beacon products include movies generated from the 256 and 512 resolution JPEG images, heliographic maps generated from a combination of the EUVI images and SDO/AIA, rotating GIF movies of the heliographic maps, and annotated images showing where any planets appear in the images. The `make_browse` script also downloads supplemental data from other missions such as SOHO and SDO. A special version of the heliographic map is generated for comparison with farside maps based on helioseismology.

3 Browse pages

The script `make_browse_last10` replaces the SECCHI JPEG images generated from the beacon telemetry with new versions based on the actual science data. At the same time, the various in situ plots are regenerated. These in situ plots are still based on the beacon telemetry packets, but use the beacon packets from the level 0 files downloaded from the MOC (see Section 4) so as to cover as complete a time range as possible.

As implied by the name, the `make_browse_last10` script steps through the last 10 days to check to see if there are any new full resolution SECCHI images that replace the original beacon

images. As each full resolution image is processed, the corresponding beacon images are moved to a `beacon` subdirectory rather than being deleted.³ Generally speaking, once all the full resolution files for a day are processed, any remaining beacon images are also moved. However, if the day still contains large gaps in the full resolution data, the beacon images within that gap are retained (except for COR1, since those beacon images aren't really useful).

After stepping through the previous 10 days, the script also steps back 27 days to support the links on each browse page that allow one to navigate by the solar rotation period. Next, the script reprocesses the coronagraph and heliospheric imager data for earlier dates to reflect the improvement in the knowledge of the instrumental background. This is done first at 15 days, and then at 35 days after the observation date. Finally, the script checks 60 and then 90 days back in time in case any new data have been added since the last time that date was processed.

The following browse image types are generated for each day:

- EUVI 171
- EUVI 195
- EUVI 284
- EUVI 304
- COR1
- COR1 running difference
- COR1 polarized brightness
- COR2
- COR2 running difference
- HI1
- HI1 running difference
- HI1 star removed
- HI2

Unlike the other products, the HI1 star removed images are mirrored from NRL, with some processing to produce the various smaller image versions. This is done through the script `ssc_update_hi_srem`.

As each date is processed, the script next checks whether the MPEG movies for that day also need to be regenerated. After all the dates have been processed, the `make_fullres` script is called to generate full-resolution heliographic maps in the `beacon` subdirectory of the website.

4 MOC data products

The script `update_moc_sds` mirrors the ancillary data products from the Mission Operations Center (MOC) STEREO Data Server (SDS), arranges the files on the SSC archive, and performs some additional processing. The files `update_moc_sds.host` and `update_moc_sds.pid` contain the host-name and process ID—these files are used to make sure that only one copy of the script is run at a time.

The first step in the process uses `rsync` to mirror the files from the SDS into a cache directory tree at `/service/stereo3/stereo/data/moc_sds`. Dated log files are kept in the directory

³This is done to address complaints from the UFO community that we were involved in a “coverup”.

`/service/stereo3/stereo/data/rsync_logs`. (Log files in this directory older than 10 days are removed.) The various data products are then copied over to the permanent archive location in the `/data/moc_sds` subdirectory of the website. Different strategies are used depending on the data product.

Many products are simply mirrored from the cache tree into the website. This is done using `rsync` based on the file extension, via the file `update_moc_sds.include`. The Mission Event Histories and Ephemeris Comparison Summaries are separately mirrored. Because the DSN Schedule Change files contain email addresses, and are of only temporary interest, they are not archived. For security reasons, the telemetry dictionary is archived outside of the website directory tree, in the directory `/service/stereo3/stereo/data/telemetry_dictionary`.

The next step in the MOC ingest process uses IDL to perform various functions. This includes generating the Archived Product Lists. A subsequent `rsync` call copies these lists back to the MOC SDS. The same routine that produces the Archived Product Lists also copies the latest telemetry files into the `/data/telemetry` website directory organized by instrument, year, and month. This directory tree also includes telemetry from the spacecraft Command and Data Handling (`cdh`) and Guidance Control (`gc`) subsystems. A separate `purge_telem` script purges older versions of the telemetry files six months after the final `.fin` versions are available.

Because some of the ancillary data products consists of a large number of files, it was decided to organize them by year. This is done through the IDL routine `ssc_ingest_ancillary.pro`, which is called separately for each data product.

The routine `ssc_ingest_stereo_kernel` compiles the SPICE ephemeris files from text transfer format to binary format, copies them to the appropriate directory within the SolarSoft tree, and registers them in a UIT-format database within IDL. Next, the routine `ssc_form_kernel_lists` writes out the text files in the SolarSoft tree that control which kernels are loaded and in what order. Finally, the routine `ssc_remove_old_ephem` removes kernels from SolarSoft that are no longer used; however, the original text transform formatted files remain in the archive.

Full resolution Attitude History files are written to `$$$WDB/stereo/gen/spice/ah`, but reduced versions of these files are written to `$$$W/stereo/gen/data/spice/ah` using a combination of IDL, the Fortran program `cksmooth`, and the SPICE utility `cksmrg`. The latter two executables are found in the `$$$SOFTWARE` directory. The procedure used produce these greatly reduced data files involves first smoothing the data, followed by spline fits.

After IDL has exited, and the Archived Product Lists have been copied to the MOC SDS, the Perl script `parse_dsn.pl` is used to generate the DSN summary page on the website.

5 In situ data and software

The script `update_ins_data` mirrors the IMPACT and PLASTIC data from the instrument sites into the SSC archive. The associated script `start_update_ins_data` is used to start the script as a cron job. The files `update_ins_data.host` and `update_ins_data.pid` contain the hostname and process ID—these files are used to make sure that only one copy of the script is run at a time.

Individual dated log files for each instrument are written to the directory `/service/stereo3/stereo/data/rsync_logs`. Log files older than 10 days are removed by the `update_moc_sds` script (see Section 4).

A similar script, `update_swaves`, mirrors S/SWAVES data from the University of Minnesota. This is currently a separate script, since the processing of S/WAVES data at UMinn is still somewhat experimental, but eventually will be merged into `update_ins_data`.

A separate script, `update_bldg21_insitu`, mirrors the in situ and radio data to the secondary “stereo.gsfc” archive.

5.1 IMPACT data and software

Most of the IMPACT data are mirrored from the `stereo.ssl.berkeley.edu` ftp site via the Perl `mirror.pl` script. The associated file `impact.mirror` controls this process. Additional Level 3 data are mirrored via `wget` from `https://stereo-dev.epss.ucla.edu/media/`.

Software routines and calibration files for processing IMPACT beacon data are mirrored into SolarSoft from the Berkeley ftp site, controlled by the file `impactsoft.mirror`. These files are written to the directory tree `/service/solarsoft` where specific parts of the SolarSoft library are mounted with write access.

5.2 PLASTIC data and software

PLASTIC data are mirrored using `rsync` from several computers located at the University of New Hampshire. One of the challenges of mirroring the PLASTIC data is combining the data from multiple locations into a single directory tree. For that reason, there are multiple `rsync` calls covering the various products. Occasionally, the PLASTIC team will ask to have older versions of certain products moved to a subdirectory, to avoid confusion with the latest version of the files. The version number in the PLASTIC file names represents the version of the software used to produce the files, not how many times the data were processed. When moving old files, it’s important to do this on both the primary archive, and also on the secondary archive on `hifi`. Otherwise, the secondary archive will end up with duplicate versions, since all mirroring is done without deletions.

`Rsync` is also used to mirror the PLASTIC software into SolarSoft. The routines in the `idl` subdirectory are used to process the PLASTIC beacon data, while the `data` subdirectory includes calibration information used by the software. PLASTIC also provides their data analysis software as zip files in the parallel `software` directory.

5.3 Status pages

The `update_ins_data` script also mirrors information about instrument data status into the `/data/status` website subdirectory. PLASTIC status information is mirrored via `rsync`, and SECCHI status information is downloaded via `wget`.

5.4 S/WAVES data

S/WAVES primary science data are mirrored from the `ftp.space.umn.edu` ftp site via the Perl `mirror.pl` script. The associated file `swaves.mirror` controls this process. Although the ftp site contains other subdirectories, only the `level-3` directory is mirrored, as the others are not useful for users, and may even be empty.

Older versions of the S/WAVES data, when they were being produced at Goddard, are stored in the subdirectory `old`, and are kept simply for historical purposes.

5.4.1 S/WAVES Level 2 data

The subdirectory `bursts` contains Type II and IV radio bursts observed by S/WAVES and WIND/WAVES. These data are presented as HTML web pages taken from the website `solar-radio.gsfc.nasa.gov/wind/data_products.html`. The facts that these are webpages, and mix together information from both the WIND and STEREO missions, make it difficult to fully automate the mirroring of these pages onto the SSC archive. However, these burst pages are not frequently updated, so updating them in a procedure which incorporates some by-hand checking is not particularly burdensome.

The script `SOFTWARE/swaves_type2/update_swaves` can be run as needed to mirror over the `solar-radio` website to a temporary directory. This script then runs an IDL program to modify the burst pages into a form suitable for the SSC website. These modified web pages are written to the subdirectory `bursts`, with images stored in `bursts/burst_images`. These pages should be already be in a form suitable for the SSC website, but this indirect approach allows some checking before the files are transferred to the SSC archive. WIND/WAVES data goes back to 1994, but only the data from the STEREO era, starting with 2006, is put on the SSC website.

6 SECCHI data

The volume of SECCHI data far outstrips those of the other STEREO instruments. For that reason, a more elaborate scheme is used to mirror the SECCHI data. The script `update_ssc` mirrors the primary SECCHI data product, which the SECCHI team refers to as “Level 0.5”. The associated script `start_update_ssc` is used to start the script as a cron job. The files `update_ssc.host` and `update_ssc.pid` contain the hostname and process ID—these files are used to make sure that only one copy of the script is run at a time.

The SECCHI “Level 0.5” data are mirrored from the NRL computer “louis14” through a sequence of `rsync` calls for each subdirectory on the NRL server. This is accomplished by stepping through the following loops:

```
spacecraft: a, b
data type: cal, img, seq
telescope: cor1, cor2, euvi, hi_1, hi_2
date: Increments 15 days backwards from today, e.g. 20210429, 20210428, ...
```

The script `ssc_make_secchi_links` creates an alternate directory tree structure where the eight-character data directories are split into separate year, month, and day directories. This alternative directory tree structure is found at `/data/ins_data/secchi/L0_YMD` on the website.

After mirroring the main “Level 0.5” files, the same procedure is used to mirror the alternate realtime (rt) and playback (pb) trees outside of the website to `/service/stereo4/data/secchi`.

After the data are mirrored from NRL, the script `start_update_bldg21` launches `update_bldg21` to mirror the SECCHI data to the secondary “stereo.gsfc” archive. This script uses a similar loop scheme as `update_ssc`, except that rather than stepping through the previous 15 days, the script steps through those date directories which have been *modified* in the last 15 days.

The script `update_ssc_date` can be used to mirror specific dates from NRL. Once a week, the script `update_ssc_year` scans over the previous year to catch any new data that wasn’t caught by `update_ssc`. Once a month, the script `update_ssc_older` scans over the entire mission.

6.1 HI Level 2 data

The script `ssc_update_hi` mirrors processed HI Level 2 data from the Rutherford Appleton Laboratory. The associated script `start_ssc_update_hi` is used to start the script as a cron job. The files `ssc_update_hi.host` and `ssc_update_hi.pid` contain the hostname and process ID—these files are used to make sure that only one copy of the script is run at a time. After mirroring the data into the primary archive, they’re also copied to the secondary archive.

These data are available on the website at `/data/ins_data/secchi_hi/` (specifically in the L2 subdirectory), though a logical link allows these data to also be seen as a subdirectory of `/data/ins_data/secchi`. The processing of the Level 2 files involves removing an approximation of the F-corona based on the mean of the lowest quartile of pixels over a running window of N days, where N is usually 1 or 11 for HI-1, and 3 or 11 for HI-2. Several different units are available: DN/s/pixel, Mean Solar Brightness, and S10. The latter two have the diffuse source correction applied.

Also in the `secchi_hi` region of the website are subdirectories (`L2_1_25` and `L2_11_25`) containing older L2 files—these directories are no longer updated. The subdirectory `L2_SWRI` contains a separate product from the Southwest Research Institute. No script actively updates these data.

6.2 EUVI wavelet images

Contrast-enhanced EUVI images using wavelet processing are generated at the Johns Hopkins University Applied Physics Laboratory, and are available at `sd-www.jhuapl.edu/secchi/wavelets`. The script `update_secchi_wavelet` mirrors the most recent data to the SSC archive, where they are made available in the `/data/ins_data/secchi/wavelets/` directory on the website. The files `update_secchi_wavelet.host` and `update_secchi_wavelet.pid` contain the hostname and process ID—these files are used to make sure that only one copy of the script is run at a time.

The first part of the script mirrors the `2_colors` directory, which contains daily movies, and is relatively small compared to the rest of the directory tree. The other two subdirectories, `fits` and `pngs`, are very large, and are organized by year and month. For each of these two subdirectories, the script steps through the previous month and the present month.

This data product is so large, it has its own pair of file systems. Most of the data are on the `/service/secchi_euvi` file system, but the 2007 data have been moved to the `/service/secchi_euvi2` file system to allow growth in the first.

7 COR1

The directory `SOFTWARE/cor1` contains software used to generate the COR1 background images. COR1 background images fall into the following categories:

Daily median images: Background images derived from a day’s worth of data. Although these are described as “median” images, they’re actually generated through a process of breaking a day down into chunks of one or more hours, generating the median in each chunk, and then the minimum of all the chunk medians. These are generated by the IDL routine `cor1_mk_daily_med.pro`. The keyword `/daily` to `secchi_prep` selects these background images.

Monthly minimum images: Monthly images generated as the minimum of the daily median images. These are generated by the IDL routine `cor1_monthly_min.pro`, and are the default background images used by `secchi_prep`.

Roll minimum images: These are minimum images generated from calibration roll data, via the IDL routine `cor1_mk_roll_min.pro`. Calibration rolls occur four times per orbit. The keyword `/raw_calroll` selects these background images.

Monthly roll images: These images are generated from a combination of the monthly minimum and calibration roll data. They are generated by the IDL routine `cor1_combine_monthly_roll.pro`, which is called from `cor1_monthly_min.pro`. The keyword `/calroll` selects these images.

Except for the roll minimum images, all the above products are generated via the script `cor1_mk_daily_med`. The calibration roll data are processed by hand. This gives one the opportunity to reject any anomalous calibration roll data, e.g. those with large spacecraft jitter. The first step in generating the calibration roll data is to run the IDL routine `setup.pro` in the `SOFTWARE/cor1` directory to direct the output to a temporary copy of the background tree. Once the new roll data have been generated and validated, the routine `cor1_make_all_combined.pro` is called to fill in the monthly roll images since the previous calibration roll.

All the COR1 background images are first written to the temporary directory tree `/service/stereo4/data/cor1/background`. The script `cor1_rsync` merges these data with the backgrounds for the other telescopes in SolarSoft and other various locations.

Although almost all the IDL routines (except `cor1_make_all_combined`) are in SolarSoft, for stability purposes the COR1 background images are generated from local copies in the `SOFTWARE/cor1` directory.

Occasionally, a new dust particle attaches itself to the COR1 objective lens, and the instrumental background undergoes a discontinuous jump. The dates and times of these jumps are maintained in the routines `cor1_monthly_min.pro` and `scc_getbkgimg.pro`. To get these modified routines into SolarSoft, they must first be copied to `louis14.nrl.navy.mil` and installed into CVS. Because it takes several days for changes made on `louis14` to propagate to SolarSoft, and just for general stability, copies of `scc_getbkgimg.pro` are kept in the `SOFTWARE/cor1`, `/beacon`, and `/scripts` subdirectories.

8 SPICE ephemerides

The SPICE ephemerides for STEREO are organized through the script `update_moc_sds` described in Section 4. The SPICE-handling software package in SolarSoft, originally written specifically for the STEREO project, has been updated to handle other missions as well. This updated package is known as SunSPICE. Additional scripts are used to keep the SPICE kernels for other missions up-to-date. This allows the positions of the STEREO spacecraft to be compared with the positions of the other supported missions.

Parker Solar Probe: The script `update_psp` in the directory `SOFTWARE/psp` on anaglyph manages the SPICE kernels for the Parker Solar Probe mission. The kernels are mirrored from the WISPR team copy at the Naval Research Laboratory. The files are first mirrored over to `SOFTWARE/psp/data`. Then the IDL routine `update_psp_ssw_data.pro` in the same directory is used to organize these files, and place them in the SSW and SSWDB trees. For certain directories which are not commonly updated, the software will send a warning mail message when new files are added to those directories, so that one can make sure the software worked correctly.

Solar Orbiter: The script `solo_update_kernels` in the directory `/Users/thompson/scripts` on moat updates the kernels for the Solar Orbiter mission. The kernels are mirrored via `wget` from the ESA ftp server to the temporary location `/Users/thompson/kernels`, after which everything is controlled through IDL routines, mostly in the same directory. First, the SolarSoft routine `ssw_wget_cleanup2` is used to clean out any files which are no longer on the ftp server. Then, the routine `solo_process_metakernels` organizes the kernels within the SSW and SSWDB trees, while `solo_purge_sunspice` removes files no longer being used. The routine `solo_restore_missing` was written to solve a problem where some files were being incorrectly purged—this problem should be solved now, but it was deemed safest to retain the call to this routine.

Solar Dynamics Observatory: The script `sdo_update_kernels` in the directory `/Users/thompson/scripts` on moat (via the IDL routine `sdo_update_kernels.pro` in the same directory) updates the file `SSW/packages/sunspice/sdo/data/sdo.bsp`.

BepiColombo: The script `ssc_update_bepicolombo` in the directory `SOFTWARE/scripts` on anaglyph calls the IDL routine `ssc_update_bepicolombo.pro` in the same directory to update the latest ephemeris for the BepiColombo Mercury Planet Orbiter (MPO) spacecraft in the directory `/service/solarsoft/sterео/gen/data/spice/other`. (Other files within this directory are updated by hand as needed.) A parallel script on moat (see Section 10.1) performs the same action for the actual `SSW/sterео/gen/data/spice/other` directory.

9 STEREO Orbit Tool

The STEREO Orbit Tool at `sterео-ssc.nascom.nasa.gov/cgi-bin/make_where_gif` is driven via an IDL server running on `stereodata`, and accessed through Remote Procedure Calls (RPC). This server is started at system boot.

The bulk of the work is done through the IDL routines `ssc_plot_where` and `sterео_coord_info`. Two scripts in `SOFTWARE/scripts` are run via cron to keep the IDL server

up-to-date and healthy. The script `load_stereo_spice` is run once a day to update the STEREO SPICE kernels, while `idlclient_retail` is run once an hour to simply send the IDL “retail” command, in case it might be stuck in the middle of a routine.

The orbit tool is accessed through the “Where is STEREO?” page at `stereo-ssc.nascom.nasa.gov/where.shtml`, which presents the same information for the present time. This page is generated through the script `make_where_is_stereo` on an hourly basis.

10 Miscellaneous scripts

This section describes a miscellaneous collection of scripts which have not yet been discussed in the previous sections. Most of these scripts run daily, epitomized by the script `make_daily` which collects a number of small daily tasks into a single script:

- Calculates ephemeris and viewperiod files to support the beacon ground stations.
- Generates in situ browse plots by Carrington rotation number.
- Determines the percentage of each days daily beacon coverage.
- Mirrors the latest IMPACT offset files to support beacon processing.
- Calls the script `remake_synoptic` to regenerate the EUVI heliographic maps for five days earlier, on the assumption that higher resolution AIA data will be available by then.

Other daily scripts include:

- `update_secchi_daily`: Calls other scripts to mirror from NRL the Cactus CME catalog, Carrington rotation maps, and polarized brightness images.
- `update_secchi_bkg`: Mirrors SECCHI instrumental background images.
- `update_secchi_synoptic_mpegs`: Mirrors SECCHI synoptic mpegs.
- `make_sos`: Generates the Science-on-a-Sphere heliographic maps in `stereo-ssc.nascom.nasa.gov/browse_sphere/`.

Conversely, the script `wget_winddata` runs every five minutes to mirror the latest 7-day solar wind data from NOAA for the beacon real-time processing.

10.1 Scripts on moat

There are also several scripts which run on the computer `moat.nascom.nasa.gov`. The reason for running these scripts on moat rather than one of the SSC computers is that moat has write access to additional parts of the SSW and SSWDB trees. These scripts are:

- `update_astron`: Mirrors the Astronomy User’s Library into `SSW/gen/idl_libs/astron`.
- `update_sumer_soft`: Mirrors SUMER software from `www2.mps.mpg.de`.

- `update_naif_earth`: Mirrors high resolution Earth PCK files used to support ITRF93 coordinates into the SunSPICE package. The underlying IDL routine, `ssc_form_naif_earth` is also run as part of the `make_daily` script (Sec. 10), but with the files written into the STEREO subtree of SolarSoft.
- `update_dscovr`: Mirrors DSCOVR solar wind files into `$$$SWDB/dscovr`.
- `mirror_stix`: Mirrors Solar Orbiter STIX software into SolarSoft.
- `update_bepicolombo`: Equivalent of `ssc_update_bepicolombo` on anaglyph, but for the actual SSW tree. See Section 8 for more information.

11 Cron jobs

The following lists the cron jobs that run on the various SSC computers at the time of writing.

On beacon:

- `restart_ssc_beacon_coverage`: Restarts `ssc_beacon_coverage` if crashed (hourly, Sec. 2.1).
- `restart_ssc_beacon_reader`: Restarts `ssc_beacon_reader` if crashed (hourly, Sec. 2.2).
- `restart_secchi_idl_ahead`: Restarts `secchi_idl_ahead` if crashed (hourly, Sec. 2.3).
- `restart_secchi_beacon_images`: Restarts `secchi_beacon_images` if crashed (hourly, Sec. 2.4).
- `/Users/thompson/scripts/startup_all`: Runs at boot time to start all the processes.

On anaglyph:

- `make_station_report`: Runs monthly to make the station report for the previous month (monthly, Sec. 2.1).
- `make_browse`: Updates today's browse page (15 minutes, Sec. 2.4).
- `make_browse_last10`: Generates the browse pages for the last 10 days (six hours, Sec. 3).
- `ssc_update_hi_srem`: Mirrors SECCHI/HI star removed images from NRL (daily, Sec. 3).
- `start_update_moc_sds`: Mirrors and processes data from Mission Operations Center STEREO Data Server (hourly, Sec. 4).
- `purge_telem`: Purges old versions of telemetry files once the final versions are available (weekly, Sec. 4).
- `start_update_ins_data`: Mirrors IMPACT and PLASTIC data (two hours, Sec. 5).
- `update_swaves`: Mirrors S/WAVES data (daily, Sec. 5.4).

- `start_update_bldg21_insitu`: Mirrors IMPACT, PLASTIC, and S/WAVES data from the primary to secondary archive (two hours, Sec. 5).
- `start_update_ssc`: Mirrors SECCHI data (two hours, Sec. 6).
- `update_ssc_year`: Scans over the last 365 days to check for any missing SECCHI data (weekly, Sec. 6).
- `update_ssc_older`: Scans over the mission lifetime to check for any missing SECCHI data (monthly, Sec. 6).
- `start_ssc_update_hi`: Mirrors SECCHI/HI data from RAL (daily, Sec. 6.1).
- `update_secchi_wavelets`: Updates SECCHI EUVI wavelet images from APL (daily, Sec. 6.2).
- `cor1_mk_daily_med`: Generates COR1 background images (daily, Sec. 7).
- `update_psp`: Updates Parker Solar Probe ephemeris files (daily, Sec. 8).
- `ssc_update_bepicolombo`: Mirrors BepiColombo ephemeris files (weekly, Sec. 8).
- `make_where_is_stereo`: Makes the “Where is STEREO?” page (hourly, Sec. 9).
- `make_daily`: Generate several products (daily, Sec. 10).
- `update_secchi_daily`: (daily, Sec. 10).
- `update_secchi_bkg`: Mirrors over the SECCHI background images from NRL (daily, Sec. 10).
- `update_secchi_synoptic_mpegs`: Updates SECCHI synoptic MPEG movies from NRL website (daily, Sec. 10).
- `make_sos`: Creates Science-on-a-Sphere heliographic maps (daily, Sec. 10).
- `wget_winddata`: Updates the latest WIND data (5 minutes, Sec. 10).
- `/Users/wtthomps/scripts/startup_all`: Runs commands needed at computer startup. At present, it appears that this is not actually needed.

On stereodata:

- `idlclient_retail`: Sends the “retail” command to the IDL RPC server (hourly, Sec. 9).
- `load_stereo_spice`: Reloads the STEREO ephemeris files in the IDL RPC server (daily, Sec. 9).

On moat:

- `solo_update_kernels`: Updates Solar Orbiter SPICE kernels (daily, Sec. 8).
- `sdo_update_kernels`: Updates the Solar Dynamics Observatory ephemeris file (daily, Sec. 8).
- `update_bepicolombo`: (daily, Sec. 8).

- `update_astron`: (weekly, Sec. 10.1).
- `update_sumer_soft`: (weekly, Sec. 10.1).
- `update_naif_earth`: (daily, Sec. 10.1).
- `update_dscovr`: (daily, Sec. 10.1).
- `mirror_stix`: (daily, Sec. 10.1).