SCF Software Programmer's Guide

Section 1: Introduction

This document is intended to be a reference for the programmers at the GLAS main SCF. It contains information about the structure of the software versioning system and procedures and tests to be performed when the software is delivered. Note: all actions are performed under the "scf" account unless otherwise specified.

Section 2: Table of Contents

Section 1: Introduction 1
Section 2: Table of Contents 2
Section 3: List of Site Names 4
Section 4: Creating Orbit Track Files 5
Section 5: How to Create Bin and Georeference Files for a New Reference Orbit 7
Section 6: Creating New Libraries 8
Section 7: Creating a New I-SIPS Release 11
Section 8: Procedures for Laser Turn-on/Turn-off and Reprocessing 12
Section 8.1: Procedures for Laser Turn-on 12
Section 8.2: Procedures for Laser Turn-off 14
Section 8.3: Procedures for Before Data Reprocessing 17
Section 8.4: Procedures for After Data Reprocessing 18
Section 9: Procedures at the Beginning of a New Year 20
Section 10: Data Transfer Software 22
Section 10.1: Creating a New Delivery Version 22
Section 10.2: Creating an Intermittent Version 24
Section 10.3: Delivering to Remote Site Procedures 25
Section 11: SCF Data Request, Processing, and Visualization Software 26
Section 11.1: Creating a New Delivery Version 26
Section 11.2: Creating an Intermittent Version 28
Section 11.3: Delivering to Remote Site Procedures 30
Section 12: Delivery Testing 32
Section 12.1: Benchmark Tests 32
Section 12.2: Remote Site Tests 33
Section 13: Standalone Tools 34
Section 13.1: Create_bn.tcl 34
Section 13.2: File_gap.tcl 35
Section 13.3: Browse_gap.tcl 36
Section 13.4: Check_subs_ea.tcl 37
Section 13.5: Check_pending_subs.tcl 39
Section 13.6: Check_pending_ea.tcl 40

Section 14: NSIDC Software 41
Section 14.1: How to Create a Tar File for the NSIDC Visualization Software 41
Section 14.2: How to Update the NSIDC IDL Product Readers 43
Section 14.3: How to Update the NSIDC F90 Product Readers 46
Section 15: The GLAS Data Visualizer 48
Section 15.1: Updating Sources for a New GSAS Version 50
Section 15.2: Creating a Visualizer Snapshot 52
Section 16: Target of Opportunity Check 54
Section 17: GLAS-SCFWEB Operation Guide 56
Section 18: Running Energy/Waveform Analysis 67
Section 19: Creating GLAS Pointing Mode Plots 71

Section 3: List of Site Names

The following are the site name abbreviations and corresponding host names for the main SCF and the I-SIPS:

- MSCF icesat0.gsfc.nasa.gov
- ISIPS isipsext.nascom.nasa.gov

The following is a list of site name abbreviations and the corresponding host names of the remote site hosts that interface with the mSCF and thus need the SCF software installed on them:

- ALT glas-scf.gsfc.nasa.gov
- BUF hudson.geology.buffalo.edu (linux)
- LIDAR glo.gsfc.nasa.gov (linux)
- LRSCF lidar-rscf.cnr.colostate.edu (linux)
- MIT mtglas2.mit.edu
- OSU parvati.mps.ohio-state.edu
- NSIDC rscf.colorado.edu (linux)
- SCFWEB glas-scfweb.gsfc.nasa.gov (linux web server)
- SIMPL simpl.gsfc.nasa.gov (linux)
- UCSD lumi.ucsd.edu
- UTCSR falcon.csr.utexas.edu (linux)
- WFF glasdev.wff.nasa.gov

The following is a list of site name abbreviations and the corresponding host names of the remote site hosts that interface with the I-SIPS but use the SCF data transfer scripts:

- CSR amundsen.csr.utexas.edu
- ISF glasist2.gsfc.nasa.gov

Section 4: Creating Orbit Track Files

Creating the orbit track files is something that is done only once for the 8-day and 91 day orbits. For GLAS, track 1 is the first track west of Greenwich.

The tracks file and the binrev and georeference files that associate with it need to be modified. These files are in directory /SCF/ancillary_data/orbit.

For 8 day repeat tracks the file names are:

Tracks file:	t8p.reforb
Binrev file:	t8preforb.CB01
Geo file:	t8preforb.CG01

For 91 day repeat tracks the file names are:

Tracks file:	t91p.reforb
Binrev file:	t91preforb.CB01
Geo file:	t91preforb.CG01

Tracks File Format:
The record length is 46412 bytes consisting of:
5801 elements of latitude array (real*4) followed by
5801 elements of longitude array (real*4) followed by
delta_time (real*4)

Start here if the first track is not starting from Greenwich (for gtrack files from UTCSR):

1. Find the number of the track that has ascending node closest to Greenwich.
2. Use /SCF/bin/ops/reorderreforb to reorder the tracks file. The source file is in /SCF/src/orbselect/working. The input file should be named reorderreforb.in5. It is a text file that includes 3 lines:
   - old tracks file name
   - new tracks file name
   - the number of the track that crosses Greenwich

Start here if the first track is starting from Greenwich (for reforb (ANC43) files from I-SIPS).

3. Run /SCF/bin/ops/mkbnrv to modify the binrev file. The source file is in /SCF/src/orbselect/working. The two arguments to run mkbnrv are:
   - binrev file name
   - tracks file name
4. Login to icesat1
5. Copy the binrev file to /de1
6. Run Geo-dir_glas with the binrev file name as an argument. (You'll get a system error in the middle of the process-ignore it!)
7. Copy the georeference file and the binrev file back to /SCF/ancillary_data/orbit. It is important that the binrev file is copied back because it is resorted by the georeference software.

Section 5: How to Create Bin and Georeference Files for a New Reference Orbit

This section gives steps to create the bin and georeference files for the track file for a new reference orbit (i.e. 91 day repeat cycle). The first two steps are performed on icesat0 and the last two steps are performed on icesat1:

<u>On icesat0:</u>

1) /SCF/bin/ops/mkbnrv <tracks file name>

2) Ftp binrev file to icesat1

<u>On icesat1:</u>

3) Geo-dir <binrev file name>

Ftp binrev file and georeference files to icesat0 /SCF/ancillary_data/orbit

Section 6: Creating New Libraries

When the GSAS releases new software with product format changes or changes to the common libraries, new libraries need to be created at the SCF. Below are the steps for creating new libraries and modules:

1. Make new versioned subdirectories (i.e V2.2) for the new libraries and modules. Create the subdirectories under /SCF/lib and /SCF/src/modules.

2. Either obtain or create the new GSAS common libraries and modules. If creating them in the ClearCase VOB, make sure that the view's config-spec is correct for making the new libraries (uses main/LATEST only). In /glas/vob/src, type "make clean", then "clearmake -V" to remake all libraries and modules.

3. Transfer the libraries to /SCF/lib/*new_version* and transfer the modules to /SCF/src/modules/*new_version*. If transferring files from the ClearCase VOB, the libraries are in /glas/vob/src/lib and the modules are in /glas/vob/src/modules. Make sure that all libraries and modules have read access for everyone.

4. From the VOB, transfer the /glas/vob/src/common_libs source code to /SCF/libsrc/*new_version.* The easiest way to do this is to type the following command: scp -r common_libs scf@icesat0:/SCF/libsrc/*new_version*

5. From the VOB, transfer the /glas/vob/data/anc07_001* files to /SCF/ancillary_data/GSAS_ANC07 (for qapg use).

6. From the VOB, transfer the /glas/vob/src/qapg to /SCF/src/qapg/working. Move the Makefile to Makefile.vob. Compare the object files listed in Makefile.vob with those in Makefile.scf. If necessary, edit Makefile.scf to account for any additional files. Copy Makefile.scf to Makefile.

7. From the VOB, transfer the files under /glas/vob/idl/qa_browse/browse, util, read, and description-files to /SCF/IDL/browse/working. To make HTML help files, invoke IDL and type <qap_description, 'qap_description.txt'>. This will create brhelp.html files that must go onto the website. To put the files on the website: on glas-scfweb go to directory /var/www/html/browser, create a directory help_<release number>, and put the files there.

8. To recompile with the new libraries:

    - cd to /SCF/bin/working
    - Edit in the Makefile: LIBVERS=library version
    - type "make clean" to remove all .o files from the subdirectories

- type “make” to create the scfcommon library and create executables. If anything did not compile go to the source directory directly and recompile. Especially check that qapg compiled because it is prone to problems.

This will recompile code in the following directories:

- /SCF/src/common/working
- /SCF/src/directories/working
- /SCF/src/data_select/working
- /SCF/src/ds_anc/working
- /SCF/src/orbselect/working
- /SCF/src/prod_create/working
- /SCF/src/qapg/working

This will also recreate the tclIndex list of all tcl files in the working directory.

9. Create “ops” directories so users exporting the SHLIB_PATH don’t have to know the explicit version number:
   - In the /SCF/lib directory, edit the version in lib_link_to_ops.ksh and run. This will link all the files in the versioned subdirectory to the “ops” subdirectory.
   - In the /SCF/src/modules directory, edit the version in mod_link_to_ops.ksh and run. This will link all the files in the versioned subdirectory to the “ops” subdirectory.

10. To change the library version in the .ksh files, edit the SHLIB_PATH keyword in /SCF/bin/working/scf_environ.ksh.

11. If the product formats have changed in the new library:

    - Edit the DATA_RELEASE keyword in /SCF/bin/working/scf_environ.ksh and run_vm_visualizer.ksh.
    - Update the NSIDC F90 Product Readers referring to the section “How to Update the NSIDC F90 Product Readers”.

12. Test the SCF software with the new libraries by submitting a request or by running files through the subscription software.

13. If the libraries work in “working” and you want to use them in the “ops” directory:

    - cd to /SCF/bin/ops
    - Repeat steps 8 – 11 above

14. If you need to compile a program individually, you will need to pass the bin version and library version to the makefile:

- cd /SCF/src/*directory/version*
- make VERS=*bin version* LIBVERS=*library version*
  (i.e. VERS=ops LIBVERS=V3.4 or make VERS=working LIBVERS=V3.4)
- If the name of the makefile is not “Makefile”, you will have to type:
  make –f *makefile_name* VERS=*bin version* LIBVERS=*library version*

Section 7: Creating a New I-SIPS Release

In the brief time period between releases, the following steps may be taken:

1. Post a message on the “Products” bulletin board on the SCF website announcing the new release. Indicate the date and time that the first data were reprocessed so people who have all their data dumped into one directory can distinguish the new release.

2. Change the DATA_RELEASE keyword in the .ksh files in /SCF/bin/ops and /SCF/bin/working. Files that need to be modified include scf_environ.ksh, run_vm_visualizer.ksh, and nsidc_environ.ksh (ops).

3. Add a directory for the browse help files on the SCF website. On glas-scfweb go to directory /var/www/html/browser, copy the directory help_<latest_release number> to help_<new_release number>.

4. In glas-scfweb:/var/www/html/status/release_table/scf_release_table.html, change the current and previous releases for the affected laser campaign.

5. In /SCF/IDL/lib/scf_visualizer/working edit gla_constants.pro to add the new release and corresponding GSAS version. Also change the DATA_RELEASE keyword in set_env_pc.pro. Either create a new snapshot or also modify the same files in the latest existing snapshot. Recreate the icesatvis_ds.sav file as well for use in IDL VM mode.

6. In /SCF/IDL/IDLreadGLAS edit gla_constants.pro to add the new release and corresponding GSAS version. Recreate the read_glas_file.sav file for use in IDL VM mode.

7. Deliver the modified files above to all the remote sites.

Section 8: Procedures for Laser Turn-on/Turn-off and Reprocessing

Section 8.1: Procedures for Laser Turn-on

The following steps can be taken in advance of the new laser campaign:

1. The following changes need to be made for the energy/waveform analysis plots on the SCF website. The files are in glas-scfweb:/var/www/html/Energy_analysis. Changes can be seen under the "Energy/Waveform Analysis Plots" button.
   - In energy_analysis.php add new links for next laser campaign.
   - In the ea and wf subdirectories, edit and run the rename_laser.ksh script to copy files from the last laser period to the new one. Change the laser campaign in the new files.

2. In glas-scfweb:/var/www/html/Product_trends/product_trends.php, add lines for the new laser time period. Also copy product_size_oldL#.php to product_size_newL#.html and replace the old laser campaign with the new one in the new file. Changes can be seen under "Product Trends Plots" on the glas-scf website.

3. In glas-scfweb:/var/www/html/Statistics/src/create_calendar_intro_2006.pm, the new laser time period and color has to be added to the subscription statistics display on the website. You can play around with colors first by modifying /var/www/html/Statistics/stat_2006/html/2006_calendar.html. This file gets overwritten however, so once you have your color, add it to the lo_colors array and add the lines for the new laser time period in /var/www/html/Statistics/src/create_calendar_intro_2006.pm. To run type run_daily_stats_perl.ksh.
   Note: Replace 2006 with the year you are currently in (long live GLAS!).

4. Update CURRENT_REFID and CURRENT_LASER in scf_environ.ksh in /SCF/bin/ops and /SCF/bin/working.

5. Update laser_periods in prod_size_plots.pro in /SCF/IDL/prod_trends/ops and /SCF/IDL/prod_trends/working.

The following steps can be taken when the ballpark start time of the new laser campaign is known:

6. In the INSTRUMENT_UPDATES database table, add a new line for the new laser turn-on. It is very important that the J2000secs indicates the exact time of laser start-fire and the use_flag is set to "Y". Files that come before the J2000secs time will be sent to /SCF/product_sets/deleted_files. The laser fire time can be obtained from Peggy Jester.

7. In /SCF/IDL/EnergyAnalysis/ops and working, add the Julian start day of the new campaign to getcampaign.pro.

The following steps are to be taken the day the laser turns on, once the start time known exactly:

8. Update the laser start time in the INSTRUMENT_UPDATE database table.

9. Update working database with INSTRUMENT_UPDATE table.

10. Update NSIDC – NSIDC_ops database with INSTRUMENT_UPDATE table.

11. In icesat.gsfc.nasa.gov:/var/www/html/index.php, comment out the “red” line and uncomment the “green” line.

12. In icesat.gsfc.nasa.gov:/var/www/html/icesat_mission_event_table.php, add a new line with the new laser turn-on information.

13. In glas-scfweb:/var/www/html/includes/scf_header.htm and scf_movie_header.htm, comment out the “red” line and uncomment the “green” line.

14. In glas-scfweb:/var/www/html/Data_tables/scf_release_table.php, add a new line with the new laser turn-on information.

15. In glas-scfweb:/var/www/html/browser, run rm_laser_off_files.pl to clean-up the browse directories during the last laser-off time period.

16. Cron jobs that need to be uncommented out (started) in the crontab list include:

    #00 16 * * * /SCF/bin/ops/run_stat_report.ksh >> /SCF/tmp/cron_statrep_errors.txt 2>&1

17. Post a message on the “Products” bulletin board on the SCF website announcing the start of the new laser campaign. Indicate the start date, time, refID, cycle, track, and release number.

Section 8.2: Procedures for Laser Turn-off

The following steps should be taken once the ballpark stop time is known:

1. In the INSTRUMENT_UPDATES database table, add a new line for the new laser turn-off. It is very important that the J2000secs indicates the exact time of laser end-fire and the use_flag is set to “N”. Files that come after the J2000secs time will be sent to /SCF/product_sets/deleted_files. The laser fire time can be obtained from Peggy Jester.

The following steps should be taken once the laser stop time is exactly known:

2. Cron jobs that need to be commented out (stopped) in the crontab list include:

   00 16 * * * /SCF/bin/ops/run_stat_report.ksh >> /SCF/tmp/cron_statrep_errors.txt 2>&1

3. In icesat.gsfc.nasa.gov:/var/www/html/index.php, comment out the “green” line and uncomment the “red” line.

4. In icesat.gsfc.nasa.gov:/var/www/html/icesat_mission_event_table.php, add a new line with the new laser turn-off information.

5. In glas-scfweb:/var/www/html/includes/scf_header.htm and scf_movie_header.htm, comment out the “green” line and uncomment the “red” line.

6. In glas-scfweb:/var/www/html/Data_tables/scf_release_table.php, add a new line with the new laser turn-off information.

7. In /SCF/IDL/EnergyAnalysis/ops and working, add the Julian end day of the new campaign to getcampaign.pro.

The following steps should be taken after the last data are received:

8. Run /SCF/bin/ops/test_check_subs_ea.ksh <laser_period> to check the subscription and energy analysis status for the laser period. If there are any “N”, “NF”, “B”, or “W” values, then processing is not complete.

9. Run SCF/bin/ops/test_check_pending_subs.ksh 7 <laser_period> to check for outstanding bad or waiting subscriptions. This script will change them to “NF” so they will be run again.

10. Run /SCF/bin/ops/test_check_pending_ea.ksh to check for outstanding pending energy analysis procedures. This script will change them to “N” so they will be run again.

11. Run /SCF/bin/ops/run_file_gap.ksh to check for gaps between the new data products. Emails will be sent to the SCF Manager if gaps have been found. Check gaps against the “Instrument Updates” table on the SCF website to see if the laser was off. If gaps exist that shouldn’t, notify Suneel at the I-SIPS for the missing data.

12. Modify and run /SCF/bin/ops/run_browse_cleanup.ksh to remove old versions of a given release and refID in /SCF/product_sets/BROWSE.

13. Run /SCF/bin/ops/run_browse_gap.ksh to check for gaps between the new browse products. An email will be sent to the SCF Manager if gaps have been found. Check missing files against data files; if the data product exists, the browse product should exist. If gaps exist that shouldn’t, notify Suneel at the I-SIPS for the missing browse products.

14. Check that all the BNA04/GRA04 files have been created in the /SCF/product_sets/current/L# directory. If so, then remove the BNL02/GRL02 files.

15. Update working database with INSTRUMENT_UPDATE and ISIPS_PRODUCT_ID tables.

16. Update NSIDC – NSIDC_ops database with INSTRUMENT_UPDATE table.

17. Update simpl – icedata and icesat5 - working databases with ISIPS_PRODUCT_ID table.

18. From /SCF/tmp move energy_analysis* and wf* files to /SCF/IDL/EnergyAnalysis/working. Run get_borg_stats.pro on the energy/waveform analysis output text files to get statistics for the Borg table.

    i.e. idl, get_borg_stats, ‘energy_analysis_l3f.txt’, ‘wfdailystats_l3f.txt’

19. Update the Borg table with the new laser period. Ask for input from Tim Urban of UTCSR.

20. Check that all TOO’s have ran. Station files are in /SCF/ancillary_data/data/stations_data and should all have the suffix .txt_chk. If not, then run /SCF/bin/ops/run_too_date.ksh.

    i.e. run_too_date.ksh <yyyy-mm-dd>

Remember that the TOO comparison code will only work if the GLA06 files are the same release as defined by the DATA_RELEASE keyword in /SCF/bin/ops/scf_environ.ksh.

21. In glas-scfweb:/var/www/html/browser, modify and run rm_old_releases.pl to remove the old release for a given year in the browse directories if necessary.

22. Check the Product Trends product size plots on the SCF website to see if any files have aberrant sizes. If so, then check for the file the /SCF/product_sets/current subdirectory. If the file is not there, then check in /SCF/tmp/plot_info.txt. If there is an acceptable reason why the file is small (file was reprocessed, laser was off), then remove the file name from plot_info.txt and run /SCF/bin/ops/run_prod_trends.ksh to regenerate the plots. Especially look for blocks of files that have abnormal sizes.

23. Post a message on the “Products” bulletin board on the SCF website announcing the completion of data processing.

24. Run energy analysis manually in /SCF/IDL/EnergyAnalysis/test70 for new laser period. Modify scfplots_ctrl180.dat for low energy campaigns to use gain cutoff of 180. Send resultant ps and txt files to David Handcock.

25. Create point mode plots as per Section 19 for new laser period.

Section 8.3: Procedures for Before Data Reprocessing

These things should be done at the start of reprocessing a campaign:

1. Before you start receiving the new data, check /SCF/product_sets/current/L# and /SCF/product_sets/previous/L# for the disk space and release numbers of the data in them.
   - If the reprocessing is to replace just the “pod/pad” value of the release (i.e. from 128 to 428) then just remove the old release data from the current directory (including ancillary files) to make space.
   - If the reprocessing is to replace the actual release (i.e. from 428 to 429) then ask John DiMarzio to see if he wants to delete the old data or move them to /SCF/product_sets/previous/L#.

2. In glas-scfweb:/var/www/html/Data_tables/scf_release_table.php, change the current and previous releases.

3. Post a message on the “Products” bulletin board on the SCF website announcing the start of reprocessing.

Section 8.4: Procedures for After Data Reprocessing

These things should be done at the end of reprocessing a campaign:

1. Check cycle and track in GLA15 file names against the “Instrument Updates” table on the SCF website to see if last files have been received.

2. Run /SCF/bin/ops/test_check_subs_ea.ksh <laser_period> to check the subscription and energy analysis status for the laser period. If there are any “N”, “NF”, “B”, or “W” values, then processing is not complete.

3. If results above for energy analysis status are not all “D” or “Y”, run /SCF/bin/ops/test_check_pending_ea.ksh. This script will change any outstanding energy analysis procedures to “N” so they will be run again.

---------- If GLA01-4 were reprocessed:

4. If all the GLA01 files are present, check that all the BNA04/GRA04 files have been created in the /SCF/product_sets/current/L# directory.

   i.e. ls GLA04* | wc
   ls BNA04* | wc
   ls BNL02* | wc

5. If the BNA04 count does not match the GLA04 count, but the BNL02 count does, then modify the directory in /SCF/utils/scripts/cp_BNL02_to_BNA04.ksh and run. This copies all BNL02/GRL02 files to BNA04/GRA04 files. Then remove the BNL02/GRL02 files.

---

6. Run /SCF/bin/ops/test_check_pending_subs.ksh 7 <laser_period> to check for outstanding bad or waiting subscriptions. This script will change them to “NF” so they will be run again.

7. Run /SCF/bin/ops/run_file_gap.ksh to check for gaps between the new data products. Emails will be sent to the SCF Manager if gaps have been found. Check gaps against the “Instrument Updates” table on the SCF website to see if the laser was off. If gaps exist that shouldn’t, notify Suneel at the I-SIPS for the missing data.

8. When all subscriptions have been fulfilled, post a message on the “Products” bulletin board on the SCF website announcing the completion of data processing.

---------- If GLA01 was reprocessed:

9. From /SCF/tmp move energy_analysis* and wf* files to /SCF/IDL/EnergyAnalysis/working. Run get_borg_stats.pro on the energy/waveform analysis output text files to get statistics for the Borg table.

   i.e. idl, get_borg_stats, ‘energy_analysis_l3f.txt’, ‘wfdailystats_l3f.txt’

10. Update the Borg table with the new laser period. Ask for input from Tim Urban of UTCSR. Save as html file and post on SCF website.

---

11. Modify and run /SCF/bin/ops/run_browse_releases.ksh to remove files in /SCF/product_sets/BROWSE of a given release and refID. Use this to remove the browse files of the old release that has been replaced.

12. Modify and run /SCF/bin/ops/run_browse_cleanup.ksh to remove old versions of a given release and refID in /SCF/product_sets/BROWSE.

13. Run /SCF/bin/ops/run_browse_gap.ksh to check for gaps between the new browse products. An email will be sent to the SCF Manager if gaps have been found. Check missing files against data files; if the data product exists, the browse product should exist. If gaps exist that shouldn’t, notify Suneel at the I-SIPS for the missing browse products.

14. In glas-scfweb:/var/www/html/browser, modify and run rm_old_releases.pl to remove the old release for a given year in the browse directories if necessary.

15. Check the Product Trends product size plots on the SCF website to see if any files have aberrant sizes. If so, then check for the file the /SCF/product_sets/current subdirectory. If the file is not there, then check in /SCF/tmp/plot_info.txt. If there is an acceptable reason why the file is small (file was reprocessed, laser was off), then remove the file name from plot_info.txt and run /SCF/bin/ops/run_prod_trends.ksh to regenerate the plots. Especially look for blocks of files that have abnormal sizes.

Section 9: Procedures at the Beginning of a New Year

At the start of a new year, the following steps need to be taken:

1. A new calendar needs to be started for the subscription statistics on the SCF website:
   - In glas-scfweb:/var/www/html/Statistics:
     - Make a new stat_<year> directory with “html” and “images” subdirectories.
     - Add a new link for the new year in statistics.php.
   - Under “src”:
     - Create daily_stats_<year>.pl from daily_stats_<old_year>.pl and modify as necessary.
     - Create the new <year>.txt file (i.e. 2006.txt) and change the year.
     - Create create_calendar_intro_<year>.pm from create_calendar_intro_<year>.pm and change the year.
     - Create create_calendar_<year>.pm from create_calendar_<year>.pm and change the year. You will also need to change the calendar structure @dom.
     - Edit the year in run_daily_stats_perl.ksh
   - Run glas-scfweb:/var/www/html/Statistics/src/run_daily_stats_perl.ksh to update new calendar.

2. Update browse products menu:
   - In glas-scfweb:/var/www/html/Browse_products/browser_main.html add the new year to the menu bar.

3. Archive and empty tables in the MySQL database if they seem too long. Tables that are usually archived include:

   - TOO_UPDATE
   - RTSCM_UPDATE
   - RTSCM_POINTING
   - CREATION
   - DISTRIBUTION
   - DISTRIBUTION_FILES
   - SUBSCRIPTION_INPUT_FILES

When archiving database tables, it is a good convention to rename them as <database_name>_ARCHIVE_<date>

i.e. CREATION_ARCHIVE_2007_01

The TOO_UPDATE, RTSCM_UPDATE, and RTSCM_POINTING tables are visible under “INSTRUMENT_UPDATES” on the SCF website. Therefore, if these are archived, new php files need to be created in glas-scfweb:/var/www/html/Instrument_updates. The instrument_updates.php file will also need to be updated with the new php file names and labels. In the “tables” subdirectory, the latest “update” html files should be moved to “archive” files. Once this is done and the database tables are archived and emptied, on icesat0 run /SCF/src/web/ops/run_istats_display.ksh to recreate the html files and then check the web display.

Note: If the DISTRIBUTION_FILES table was archived, then in the process_subs.tcl file in /SCF/bin/ops and working, the following lines need to have the last number incremented (search on “ql.”):

```
#If data was expedited, add "ql" to filename
 if {$qli == 1} then {
   set output_str "$reqID\_$arel\_$laser_dir\ql.$pid_ext\_01_40"
 } else {
   set output_str "$reqID\_$arel\_$laser_dir.$pid_ext\_01_40"
 }
```

In this example, the “40” should be changed to “50”.

4. Backup the MySQL icesat0 ops database to the NISDC mscf_ops_backup database.

5. Backup the MySQL NSIDC NISDC_ops database to the icesat0 ops nsidc_ops_backup database.

Section 10: Data Transfer Software

Section 10.1: Creating a New Delivery Version

The data transfer software is delivered to the following remote sites: ALT, BUF, LIDAR, MIT, NSIDC, OSU, SCFWEB, UCSD, UTCSR, WFF, LRSCF, SIMPL, CSR, and ISF. The data transfer software under /SCF/src/perl is unique to each site and must be installed separately from the rest of the SCF software. Software is developed under /SCF/src/perl/working and run operationally under /SCF/src/perl/ops. Whenever a new software release is delivered (i.e. 200304.0), it must be created from "working".

- Perl scripts are linked to /SCF/src/perl/ops from a versioned subdirectory (i.e. /SCF/bin/200304.0)
- Korn shell files exist in /SCF/src/perl/ops and /SCF/src/perl/working ONLY.

The directories under /SCF/src/perl contain all files to be used at the mSCF and rSCF's. Some of the scripts are specialized for mSCF use only and are run from /SCF/src/perl/ops. In the rSCF subdirectories are contained only the files to be sent to each remote site. The scripts to the 11 rSCF's are essentially the same, except the ones for the linux systems may be slightly different from those for the HP systems. Some scripts are added for SCFWEB. ISF and CSR push and pull from the I-SIPS directly so they use additional scripts. Under the rSCF directories are contained versioned subdirectories and an "ops" subdirectory where the unique korn shell scripts exist and where the perl scripts of the delivered version are linked to.

To create the new versioned subdirectories and link files to ops:

1. Go to the /SCF/src/perl directory on icesat0

2. Edit the old and new versions in the create_new_perl_version.ksh script and run. This will:

   - Create a new versioned subdirectory under /SCF/src/perl and copy perl files from the "working" subdirectory to the new versioned subdirectory.
   - Create new versioned subdirectories under /SCF/src/perl/*rSCF* and copy perl files from the old versioned subdirectories to the new versioned subdirectories.
   - Replace all perl files in each /SCF/src/perl/*rSCF/newversion* subdirectory with the file from /SCF/src/perl/*newversion.*
   - Link perl files in new versioned subdirectories to "ops" subdirectories

3. If anything in the korn shell scripts has changed between working and ops, those changes need to be manually done in the same korn shell scripts under ops.

4. Since the .ksh scripts are unique for each site, visually compare the .ksh files in the ops subdirectory for a remote site with those in /SCF/src/perl/ops to see if any changes need to be made. If so, make them in the .ksh script for each site.

5. For SCFWEB, NSIDC, BUF, UTCSR, LRSCF, SIMPL, ISF, and CSR: In the versioned subdirectory, modify the perl scripts according to the README file.

6. Periodically you may want to remove old directories. To do this, edit the version in /SCF/src/perl/perl_rmdir.ksh and run. This will remove those versioned subdirectories under /SCF/src/perl and each /SCF/src/perl/*rscf* subdirectory.

Section 10.2: Creating an Intermittent Version

To make an intermittent delivery for the mSCF only follow these steps:

1. Go to the /SCF/src/perl directory on icesat0

2. Create a new subversioned directory (i.e. 200304.1).

3. Copy the perl scripts from /SCF/src/perl/working to /SCF/src/perl/*new_version*.

4. Link perl files from /SCF/src/perl/*newversion* to /SCF/src/perl/ops by editing the directory version in /SCF/src/perl/perl_link_to_ops.ksh and running.

5. If anything in the korn shell scripts has changed between working and ops, those changes need to be manually done in the same korn shell scripts under ops.

Section 10.3: Delivering to Remote Site Procedures

To deliver the data transfer software to the remote sites:

1. Edit the version in /SCF/src/perl/scp_perl_sites.ksh and run. You will have to type in passwords for ISF and CSR. This will put the new versioned subdirectory on all remote sites. If any .ksh files in "ops" have changed, do this for "ops" as well (get approval from Peggy and Charles before doing this on ISF and CSR, respectively). You can use /SCF/src/perl/scp_file.ksh for this.

2. For ISF: Email Peggy Jester at jester@osb.wff.nasa.gov to inform her that new data transfer scripts in ~/src/perl/*newversion* need to be linked to ~/src/perl/ops. Ask her for a good time to do so.

3. For CSR: Email Charles Webb at cewebb@mail.utexas.edu to inform him that new data transfer scripts in ~/src/perl/*newversion* need to be linked to ~/src/perl/ops. Ask him for a good time to do so.

4. Run /SCF/utils/ssh_perl_sites.ksh. You will have to type in passwords for ISF and CSR – don't do this until Peggy and Charles give approval. On each site edit the version in ~/src/perl/perl_link_to_ops.ksh and run to link perl files from ~/src/perl/*newversion* to ~/src/perl/ops. Delete the next to last version under ~/src/perl.

5. Test by transferring files to each site, checking that they are received in the correct directory on the remote site, and that the files are removed from the distribution caches on icesat0 because successful PAN's were received. Wait until 2 cron cycles pass then check for the data on the remote hosts and in the distribution caches.

6. Since the ISF and CSR communicate with the I-SIPS and not the mSCF, testing will have to be done between isipsext and the ISF and CSR sites. Note that testing on isipsext has to be done in the isipspr1:/exttest/external directory and the cron job running test_scripts.ksh has to be set.

7. To test SCFWEB, wait until browse products are transferred and check the SCF website at http://glas-scfweb.gsfc.nasa.gov to see if the browse products were updated. You can also ssh into these sites and view the /SCF/tmp/cron.log.### files for errors.

Section 11: SCF Data Request, Processing, and Visualization Software

Section 11.1: Creating a New Delivery Version

SCF software development is done under the “working” subdirectories under /SCF/src, /SCF/IDL, and /SCF/bin. Whenever a new SCF software release is delivered (i.e. 200304.0), it must be created from “working”.

The operational SCF software is kept under the “ops” subdirectories under /SCF/IDL and /SCF/bin:

- Tcl and perl scripts are linked to /SCF/bin/ops from a versioned subdirectory (i.e. /SCF/bin/200304.0)
- Korn shell and make files exist in /SCF/bin/ops and /SCF/bin/working ONLY. Therefore, the Fortran 90 files can only be complied in these directories.
- If tcl files have been added, a new tclIndex library has to be created. To recreate this list in a versioned directory (i.e. /SCF/bin/200304.0), type “make –f Makefile_tclIndex”.

To create the new versioned subdirectories and link files to ops:

1. Edit the old and new directory versions in /SCF/src/create_new_version.ksh and run. This will create new versioned subdirectories under:

    a. /SCF/IDL/data_selection
    b. /SCF/IDL/browse
    c. /SCF/src/common
    d. /SCF/src/directories
    e. /SCF/src/data_select
    f. /SCF/src/ds_anc
    g. /SCF/src/orbselect
    h. /SCF/src/prod_create
    i. /SCF/src/qapg
    j. /SCF/bin

It will populate the new versioned subdirectories with files from the old versioned subdirectories, change the group to “users” and chmod everything to 775.

2. In the /SCF/bin directory edit the version in link_to_ops.ksh and run. This will link all the files in the versioned subdirectory to the "ops" subdirectory.

3. In the /SCF/IDL/data_selection directory edit the version in ds_link_to_ops.ksh and run. This will link all the files in the versioned subdirectory to the "ops" subdirectory.

4. In the /SCF/IDL/browse directory edit the version in browse_link_to_ops.ksh and run. This will link all the files in the versioned subdirectory to the "ops" subdirectory.

5. To recompile the Fortran 90 with the new software version:
    - cd to /SCF/bin/ops
    - In the Makefile edit the src version and the library version.
    - type "make clean" to remove all .o files from the subdirectories
    - type "make" to create the scfcommon library and create executables. If anything did not compile go to the source directory directly and recompile.

This will also recreate the tclIndex list of all tcl files in the ops directory.

6. If the libraries have changed between working and ops, refer to the "Creating New Libraries" section.

7. If anything in the korn shell scripts has changed between working and ops, those changes need to be manually done in the same korn shell scripts under ops.

8. To use a new visualizer snapshot, edit the SCF_VISUALIZER environment variable in /SCF/bin/*version/*scf_environ.ksh and run_vm_visualizer.ksh.

9. Link the latest snapshot to /SCF/IDL/lib/scf_visualizer/ops by typing the following while in /SCF/IDL/lib/scf_visualizer:

   ln –s *latest_snapshot* ops

10. Test the SCF software with the new software version by submitting a request and/or by running files through the subscription software. Beware: If submitting a request, limit the time span or limit the tracks. Once submitted, check /SCF/dist/MSCF to see if the requested data are there.

11. Test the visualization software on icesat0 as per the "Benchmark Tests" section.

    Once these steps are complete, the new software version may be delivered to the remote sites as per section 9.3.

Section 11.2: Creating an Intermittent Version

SCF software developed under “working” may be copied to a versioned subdirectory (i.e. 200304.1) and then linked to “ops” once tested.

1. Create the new versioned subdirectories: How to do this depends on what you need done:

   A) If you want a new subdirectory under all source and IDL directories and /SCF/bin, edit the old and new directory versions in /SCF/src/create_new_version.ksh and run. This will create new versioned subdirectories under:

   a. /SCF/IDL/data_selection
   b. /SCF/IDL/browse
   c. /SCF/src/common
   d. /SCF/src/directories
   e. /SCF/src/data_select
   f. /SCF/src/ds_anc
   g. /SCF/src/orbselect
   h. /SCF/src/prod_create
   i. /SCF/src/qapg
   j. /SCF/bin

   B) If you want a new subdirectory under all source directories and /SCF/IDL/data_selection only, edit the old and new directory versions in /SCF/src/create_new_f90_version.ksh and run. This will create new versioned subdirectories just under:

   a. /SCF/IDL/data_selection
   b. /SCF/src/common
   c. /SCF/src/directories
   d. /SCF/src/data_select
   e. /SCF/src/ds_anc
   f. /SCF/src/orbselect
   g. /SCF/src/prod_create
   h. /SCF/src/qapg

C) If you just modified a tcl script, make the versioned subdirectory under /SCF/bin and copy the .tcl and .pl scripts from /SCF/bin/working to /SCF/bin/*new_version*.

2. In the /SCF/bin directory edit the version in link_to_ops.ksh and run. This will link all the files in the versioned subdirectory to the "ops" subdirectory.

3. In the /SCF/IDL/data_selection directory edit the version in ds_link_to_ops.ksh and run. This will link all the files in the versioned subdirectory to the "ops" subdirectory.

4. In the /SCF/IDL/browse directory edit the version in browse_link_to_ops.ksh and run. This will link all the files in the versioned subdirectory to the "ops" subdirectory.

5. To recompile the Fortran 90 with the new software version:

   - cd to /SCF/bin/ops
   - In the Makefile edit the src version and the library version.
   - type "make clean" to remove all .o files from the subdirectories
   - type "make" to create the scfcommon library and create executables. If anything did not compile go to the source directory directly and recompile.

6. If the libraries have changed between working and ops, refer to the "Creating New Libraries" section.

7. If anything in the korn shell scripts has changed between working and ops, those changes need to be manually done in the same korn shell scripts under ops.

8. When testing special requests, send the request email to the following addresses to test the correlating software:

   scf@icesat0.gsfc.nasa.gov to use software in /SCF/bin/ops
   scftest2@icesat0.gsfc.nasa.gov to use software in /SCF/bin/working

Section 11.3: Delivering to Remote Site Procedures

The data request and data visualization software is delivered to all remote sites, HP and Linux, however the methods are slightly different since the Fortran files, including common libraries, have to compiled on each Linux system separately.

Once the new delivery version has been created, the following steps need to be taken to deliver the software to the remote sites:

Section 1: Test the data request and visualization software on icesat0 under an account other than "scf" (i.e. scftest), as per the "Benchmark Tests" section.

Section 2: Install the software on the linux test system, icesat6. This is done with a tar file. To create and install the tar file:

a. Go to the /SCF/dist/tar_files directory on icesat0.
b. Create a new subdirectory named with the version and copy and rename the make_tarfile script from the previous version's subdirectory.
c. Edit the make_tarfile script to reflect the new tar file name and the directories and files that need to be tarred.
d. Run the make_tarfile script to create the tar file.
e. Scp the tar file to icesat6. I.e. scp tarfile icesat6:/SCF
f. Ssh into icesat6:
    - From the /SCF directory (where the tar file was transferred to), type "tar xvf tarfile" to extract the tar file
    - Check to ensure that the files are there
    - Check the permissions in the /SCF/IDL/lib/scf_visualizer/ops directory. All files should be readable by all.
    - Check the permissions in the /SCF/IDL/help directory. All files should be readable by all.
    - Follow the Instructions in the "SCF Linux Installation Guide" to install the software on the Linux sites

Section 3: Test the data request and visualization software on icesat6.

Section 4: Install the software on the remaining remote sites. To install the same tar file as was installed on icesat6:

a. Go to the /SCF/dist/tar_files directory on icesat0
b. Edit the version and file in the /SCF/dist/tar_files/scp_tar_file.ksh script. Run to transfer the tar file to each remote site.
c. Run /SCF/utils/ssh_rscf_sites.ksh and follow the steps above to untar the tar file on each site.

Section 5: Follow the Instructions in the "SCF Linux Installation Guide" to install the software on the Linux sites.

Section 6: Periodically you may want to remove old libraries and src directories. To help do this, edit the version in /SCF/src/src_rmdir.ksh and run. This will remove those versioned subdirectories under /SCF/src, /SCF/IDL, and /SCF/bin. Also:

- Remove old libraries under /SCF/lib and old modules under /SCF/src/modules.
- Remove old data transfer scripts under /SCF/src/perl.
- Remove old IDL visualizer snapshots under /SCF/IDL/lib/scf_visualizer (not on icesat0)
- Check /SCF/tmp to make sure that the cleanup cron is removing old dir_* directories.

Section 7: Test the data request and visualization software on the remote sites using the "scftest" account, as per the "Benchmark Tests" section.

Section 8: Delete the test and log files from all remote sites.

Section 9: Update documentation as necessary and put new pdf files on the SCF website.

Section 10: Update the release notes for this software delivery.

Section 11: Email the GLAS science team to inform them that a new version of the software has been delivered and attach the release notes.

Section 12: Delivery Testing

Section 12.1: Benchmark Tests

The following tests should be performed on glas-scf under the “scftest” account before a new visualizer snapshot is installed on all the remote sites:

- Read the SNAPSHOT_NOTES in the /SCF/IDL/lib/scf_visualizer/snapshot… directory that you’re testing to see what has been modified so you can specifically test those changes
- Launch all plotsets for every product, even if you don’t look at them all, just to make sure that they work
- Do at least one altimetry and one lidar exercise in the SCF Visualizer Tutorial

Section 12.2: Remote Site Tests

The following tests should be performed on each remote site under the “scftest” account after an SCF software delivery is installed:

- Invoke the data request software from the home directory using /SCF/bin/*new_version*/run_data_request.ksh
- Select special request
- Have requested data go into a special test directory
- Select an altimetry product and a lidar product
- Check that help files can be accessed from data request software
- Save the parameter file
- Check that emails were received
- Ensure that files were transferred and removed from mSCF
- Check that log file was received in /SCF/product_sets/logs even if no data was received.
- Invoke the visualization software from the home directory using /SCF/bin/*new_version/*run_visualizer.ksh
- View browse products
- Visualize the altimetry and lidar products
- Check that the tracks received were the tracks requested
- Check that print options work in all types of windows
- Check that data output works in all types of windows
- Check that help files can be accessed from visualization software

Section 13: Standalone Tools

Section 13.1: Create_bn.tcl

General Description:
Creates BN and GR files all GLA files in input directory

Execution:
run_create_bn.ksh <data_directory>

Example:
- run_create_bn.ksh /SCF/product_sets/current/L2E

Section 13.2: File_gap.tcl

General Description:
The script file_gap.tcl searches for gaps in file names in the specified input directory. It also checks against the start and end refID/cycles/tracks in the database to find gaps at the start and end of campaigns. It can be run from /SCF/bin/ops.

Execution:
run_file_gap.ksh

Output:
file_gap.txt. This file is also emailed to the SCF manager.

Example of an output:

```
/SCF/bin/working/file_gap.txt
The following are gaps in the list of file names:

dir=/SCF/product_sets/current/L2B

Time gap between GLA01_029_2107_001_1310_2_01_0001.P0374 and GLA01_029_2107_001_1317_1_01_0001.P0375
Gap start date/time: 2004-02-19 16:13:14  Gap end date/time: 2004-02-20 02:41:32

Time gap between GLA01_029_2107_002_0268_4_01_0001.P0397 and last track 421
Start date/time: 2004-03-11 15:09:54  End date/time: 2004-03-21 20:38:46
```

Meaning:
In the /SCF/product_sets/current/L3B directory, there is a gap in GLA01 files from reference ID=2107, cycle=1, track=1310, segment=2 to track=1317, segment 1. That means that track 1310, segments 3 and 4 are missing, as well as all segments for tracks 1311-1316. There is also a gap for GLA01 from cycle 2, track 269 to last track 421.

What to Do:
First check that there really is a gap between files and determine which files are missing. Then check the reference orbit, cycle, and tracks of the gaps against the Instrument Updates - Instrument Updates table on the SCF website. Some laser time periods have inherent gaps due to mechanical problems (i.e. in the above example, tracks1310 to 1316 are missing because the satellite entered a sun acquisition mode). If the gaps are present when the use_flag on the table =’Y” then request the missing data from the I-SIPS.

Section 13.3: Browse_gap.tcl

General Description:
The script browse_gap.tcl searches for gaps in file names for a specific release and refID in the /SCF/product_sets/BROWSE directory. It can be run from /SCF/bin/ops. Note: this script does search for gaps down to the segment level. However, it only searches on the first browse file in a set, png.0.

Execution:
run_browse_gap.ksh

Output:
browse_gap.txt. This file is also emailed to the SCF manager.

Example of an output:

```
/SCF/bin/ops/browse_gap.txt
The following are gaps in the list of file names:
release=28
refid=1102
Time gap between GLA01_028_1102_005_0095_4_01_BRWS_0001.png.0 and
GLA01_028_1102_005_0098_2_01_BRWS_0001.png.0
```

Meaning:
In the /SCF/product_sets/BROWSE directory, there is a gap in GLA01 files from reference ID=1102, cycle=5, track=95, segment=4 to track=98, segment 2. That means that tracks 96 to 98, segment 1 are missing.

What to Do:
First check that there really is a gap between files and determine which files are missing. Then check the reference orbit, cycle, and tracks of the gaps against the Instrument Updates - Instrument Updates table on the SCF website. Some laser time periods have inherent gaps due to mechanical problems. If the gaps are present when the use_flag on the table =’Y” then request the missing data from the I-SIPS.

Section 13.4: Check_subs_ea.tcl

General Description:
The script check_subs_ea.tcl checks subscriptions and energy analysis status in database for input laser period

Execution:
test_check_subs_ea.ksh <laser_period>

Output:
Prints to screen.

Example of an output:

```
laser=L3I
Subscription Status
B, 39
D, 486
N, 5030
P1, 40
P2, 56
W, 125
Y, 1548

Energy Analysis Status
D, 5218
Y, 1981
errorCode=0
```

Meaning:
For subscriptions for laser period L3I, there are 1548 files that completed processing, 96 files in process, 5030 files that need processing, 486 files that don't need processing, 39 files that bombed processing, and 25 files waiting until the 14 rev product set is complete to be processed.

For energy analysis for laser period L3I, there are 1981 files that completed processing and 5218 files that don't need processing.

What to Do:

If there are any “N” files (not processed) then just wait until they are processed. Track their progress to ensure that the process_subs.tcl process is running. Also, you can type, “ps –ef |grep process_subs” to check that the process is running. There should be one or two processes running. “D” files are GLA02-3 files and since they are not stored, they cannot have subscriptions fulfilled with them. “B” files are usually GLA04 files that have to be reprocessed once all the files are complete so the BNL02 files will be available. Once there are no “N” files, the script, check_pending_subs.tcl can be run to set to “B” files to “N” for reprocessing. It there are still “W” files waiting to be processed and the data have stopped coming, then run check_pending_subs.tcl to set to “W” files to “NF” for reprocessing.

Section 13.5: Check_pending_subs.tcl

General Description:
The script check_pending_subs.tcl checks for bad, pending, or waiting files in database and sets them to "N" or “NF” (depending on reset option) forcing them to have subscriptions run against them.

Execution:
test_check_pending_subs.ksh <reset option> <laser period>

Reset options:
- Call with no input arguments to change P1, P2, P3, P4 and B to N
- Call with input arguments 1, 2, 3, or 4 to change P1, P2, P3, or P4 to N
- Call with input arguments 6, 7 to change W or B to N or to NF

Output:
Prints to screen.

Example of an output:

```
Subscriptions are 'B' for the following files and will be changed to 'N'
file=GLA05_428_2123_002_0309_2_01_0001.P1848
file=GLA05_428_2123_002_0309_3_01_0001.P1848
file=GLA06_428_2123_002_0309_1_01_0001.P1848
file=GLA06_428_2123_002_0309_3_01_0001.P1848
errorCode=0
```

Meaning:
Four files were found to have status “B” meaning that they failed processing for some reason. Their status has been changed to “N” so processing on them will be tried again.

What to Do:
Just let the cron come in and try to reprocess the files again. Keep an eye out of emails indicating errors. If none appear, the run test_check_subs_ea.ksh again to see the status of the files.

Section 13.6: Check_pending_ea.tcl

General Description:
The script check_pending_ea.tcl checks for bad or pending energy analysis files in database and sets them to "N".

Execution:
test_check_pending_ea.ksh

Output:
Prints to screen.

Example of an output:

```
Energy analysis are 'P' for the following files and will be changed to 'N'
file=GLA01_028_2123_002_0309_2_02_0001.P1848
file=GLA01_028_2123_002_0309_3_02_0001.P1848
errorCode=0
```

Meaning:
Two files were found to have status “P” meaning that the processing failed for some reason. This is only true if the energy analysis software is currently not running, if it failed or is taking an unreasonably long time, appearing to be hung. Their status has been changed to “N” so processing on them will be tried again.

What to Do:
Just let the cron come in and try to reprocess the files again. Keep an eye out of emails indicating errors. If none appear, the run test_check_subs_ea.ksh again to see the status of the files.

Section 14: NSIDC Software

Section 14.1: How to Create a Tar File for the NSIDC Visualization Software

This section gives steps to create the tar file for the NSIDC visualization software given to the public for use in Unix and PC environments. It includes a test data set and documentation.

Steps in building the distribution software:

1. Cd to **/SCF/IDL/lib/idl_visualizer.**

Below is the directory tree**:**

- **README**
- **SCF_IDL_DVUG.doc**
- **anc/**
- **color_table/**
- **common/**
- **data/**
- **data_selection/**
- **dem/**
- **help/**
- **icesatvis_ds.pro**
- **icesatvis_ds.sav**
- **images/**
- **make_IDL_tarfile.ksh**
- **make_save_file.ksh**
- **make_save_file.tcl**
- **map/**
- **orbit/**
- **parameters/**
- **revfile/**
- **run_ds**
- **run_visualizer.ksh**
- **run_vm_visualizer.ksh**

- **scf_visualizer/**
- **set_env_pc.pro**
- **stations/**

2. Update the **SCF_IDL_DVUG.doc**.

3. Remove the old snapshot directory under **/SCF/IDL/lib/idl_visualizer/scf_visualizer**.

4. Copy the latest snapshot directory under **/SCF/IDL/lib/scf_visualizer** to **/SCF/IDL/lib/idl_visualizer/scf_visualizer**.

5. Update the following environmental variables in **run_visualizer.ksh, run_vm_visualizer.ksh, and set_env_pc.pro:** SCF_VISUALIZER and DATA_RELEASE

6. Update the visualizer snapshot in **make_save_file.ksh** and **make_IDL_tarfile.ksh**

7. To create the IDL virtual machine save file, type **make_save_file.ksh.** This copies files from the data_selection directory to the new snapshot directory and creates **/SCF/IDL/lib/idl_visualizer/icesatvis_ds.sav**.

8. Test the visualizer by running: **run_visualizer.ksh.** Especially check that any new modifications work.

9. Test the visualizer in the IDL virtual machine mode by running: **run_vm_visualizer.ksh.**

10. Run the script **make_IDL_tarfile.ksh** to tar and gzip the distribution.

11. Scp the tar file just created to the hard drive on the PC. Once there, ungzip and untar the file with the IZArc or other utility. You should see the same directory tree as mentioned above.

12. Test the visualizer on the PC under the IDL virtual machine as follows:

    - Make sure you have the IDL virtual machine (VM) installed. If not, you can download it for free at http://www.ittvis.com/ProductsServices/IDL/IDLModules/IDLVirtualMachine.aspx
    - Browse to get to the **icesatvis_ds.sav** file and double-click to open. Make sure that you select the IDL binary file and not the IDL source file, icesatvis_ds.pro.
    - Click on the IDL virtual machine flash screen.

13. Once testing is satisfactorily completed, scp the tar file to NSIDC or other sites as needed.

Section 14.2: How to Update the NSIDC IDL Product Readers

The IDL product readers for delivery to NSIDC print only the product variables; they do not convert to algorithm variables like the visualizer does. The product structures need to be updated with every product release. The GLA01 waveform reader is included in this write-up.

NOTE: The following steps will create new product structure (glaNN_pvV_V__define.pro) files for all products regardless of whether the product structure actually changed in the new version for any particular product. This is to keep the code consistent and easy to understand and maintain.

To update the product structures:

1. Go to the /SCF/IDL/IDLreadGLAS directory on icesat0

2. Edit the old and new versions in the cp_version_pro.ksh script and run. This will copy define.pro files for all products with the old version in the name to files with the new version in the name.

3. Edit the old and new versions in the change_version_pro.ksh script and run. This will change the old versions to the new versions within the define.pro files for all products.

4. Copy /SCF/IDL/lib/scf_visualizer/working/gla_constants.pro to this directory.

5. Copy /SCF/IDL/lib/scf_visualizer/working/*pvV_V__define.pro to this directory.

6. Copy /SCF/IDL/lib/scf_visualizer/working/unpack_alt_flags_vV_V.pro to this directory.

7. In read_glas_file.pro add a line for the new GSAS version to each case statement.

8. In read_gla01_wf.pro add a line for the new GSAS version to each case statement.

9. If flags have changed (refer to the GSAS delivery release notes), you'll have to update the algorithm structure, print statements, and Prod2Alg routine in the affected glaNN_uflags_vV_V_prod2alg.pro file.

10. Change the product structure version (search on check_version) in too_gla06_check.pro, glas_compare.pro, and high_dem_ret.pro.

11. Do the following to enable the readers to be run under the IDL virtual machine:

    11.1: In compile_read_glas_file.pro add the new version files to the list.
    11.2: Create the read_glas_file.sav file by typing at the IDL prompt: @compile_read_glas_file.pro

12. Do the following to enable the waveform reader to be run under the IDL virtual machine:

    12.1: In compile_read_gla01_wf.pro add the new version files to the list.
    12.2: Create the read_gla01_wf.sav file by typing at the IDL prompt: @compile_read_gla01_wf.pro

13. Do the following to enable the SRTM reader to be run under the IDL virtual machine:

    13.1: In compile_high_dem_ret.pro add the new version files to the list.
    13.2: Create the high_dem_ret.sav file by typing at the IDL prompt: @compile_high_dem_ret.pro

14. Test the changes to the readers on all products. Edit the control file, read_glas_ctrl.dat, to change the product name or number of records to print. The readers can be run in one of two ways:

    14.1: IDL interactive: Invoke IDL first and type: read_glas_file
    14.2: IDL virtual machine: Type at the Unix command line: idl -vm='read_glas_file.sav'. Click on the splash screen when it comes up.

15. Test the changes to the GLA01 waveform reader. Edit the control file, read_gla01_ctrl.dat, to change the product name or number of records to print. The reader can be run in one of two ways:

    15.1: IDL interactive: Invoke IDL first and type: read_gla01_wf
    15.2: IDL virtual machine: Type at the Unix command line: idl -vm='read_gla01_wf.sav'. Click on the splash screen when it comes up.

16. Test the changes to the SRTM reader. Data for testing high_dem_ret are in SCF/product_sets/SRTM_test. Copy high_dem_ret.sav to this directory. Edit the control file, high_dem_ret_ctrl.dat, to change the product name. To run the reader:

    16.1: IDL virtual machine: Type at the Unix command line: idl -vm='high_dem_ret.sav'. Click on the splash screen when it comes up. The output file, GLA06_IDL_DEM.dat, is automatically created.

17. Transfer the new code to the NSIDC system, rscf. To do this type:

    scp * rscf:/SCF/IDL/IDLreadGLAS (This assumes that you are logged into icesat0 as user scf.)

18. Contact Matt Savoie, Dave Korn, and Doug Fowler at NSIDC when the new software has been delivered. You may also need to notify John DiMarzio and David Hancock so they can include the software in the official DAP to NSIDC.

Section 14.3: How to Update the NSIDC F90 Product Readers

The Fortran 90 product readers for delivery to NSIDC print the product variables; they do not convert to algorithm variables and flag values are printed in hex format. The product structures need to be updated with the new library routines for every product release.

To update the product structures:

1. Go to the /SCF/src/readGLAS directory on icesat0.

2. Make a new versioned subdirectory (i.e V5.0) under this directory.

3. Copy the contents of the last versioned subdirectory to this new subdirectory.

4. New libraries should have been created for the same version as this new subdirectory. If not, refer to the "Creating New Libraries" section. Go to the /SCF/libsrc/*version*/common_libs/prod_lib directory. For each product type (xx) copy the following files from the libsrc subdirectory to the readGLAS subdirectory:

   - GLAxx_hdr_mod.f90
   - GLAxx_mod.f90
   - GLAxx_print_mod.f90
   - GLAxx_prod_mod.f90

   Note: Remove GLA00 files

   Also need for GLA06:
   - GLA06_alg_mod.f90
   - GLA06_flags_mod.f90
   - GLA06_scal_mod.f90

   and
   - common_flags_mod.f90
   - prod_def_mod.f90

5. Type "make" to compile.

6. Test the changes on all products. Edit the example test file, test_example.txt, to change the product names or number of records to print. To run, copy each line from the file onto the command line.

7. Also need to test read_gla01_wf (GLA01) and high_dem_ret (GLA06). Data for testing high_dem_ret are in /SCF/product_sets/SRTM_test.

8. Transfer the new code to the NSIDC system, rscf. To do this from the /SCF/src/readGLAS directory type:

   scp –r *versioned_subdirectory* rscf:/SCF/src/readGLAS (This assumes that you are logged into icesat0 as user scf.)

9. Contact Matt Savoie, Dave Korn, and Doug Fowler at NSIDC when the new software has been delivered. You may also need to notify John DiMarzio and David Hancock so they can include the software in the official DAP to NSIDC.

Section 15: The GLAS Data Visualizer

Development of the ICESat/GLAS Data Visualizer takes place primarily in the directory **/SCF/IDL/lib/scf_visualizer/working/**. However, there are also parts in several other directories:

/SCF/IDL/lib/common/ -- Non-SCF-specific source files, third-party library routines, etc.

/SCF/IDL/lib/product_readers/ -- The IDL product and algorithm structure definitions, product-to-algorithm converters, product reader classes, and their supporting source files live here. Note that a new GSAS version will usually not change every product, so we don't necessarily create new versions of these files for each GSAS release. The reader classes and such contain CASE statements that choose the most recent definitions up to the version or release specified.

/SCF/IDL/lib/utilities/ -- Utility programs that are not strictly speaking part of the Visualizer, but which are included with it and may make use of Visualizer routines. For example, there are programs to dump request and UR files here.

/SCF/IDL/lib/working/help/ -- Visualizer help text files.

In addition, a few files (mainly front-end routines that are needed by the VM version) are linked from elsewhere in the directory tree.

The files in these directories (except help/) all have symbolic links in scf_visualizer/working/, so that IDL can easily locate them without needing to have each directory added to !path. These files should all be edited in their "home" directories; it is best to not edit the symbolic link itself. If other files that are to be part of the Visualizer are added to these directories, be sure to create the symbolic link (**ln –s *target link***) in working/.

When editing Visualizer source files, remember to add appropriate comments to the HISTORY section of the file header!

When appropriate, a snapshot of the Visualizer source code is created in a new directory under /SCF/IDL/lib/scf_visualizer/ (see section 17.1). Most snapshots are named **snapshot_YYYYMMDD**, where YYYYMMDD is the date. Appropriate times to create a snapshot include freezing the sources for a release; when beginning a major change that could leave working/ unstable for an extended period; when a major new feature has been completed; or simply when enough smaller changes have accumulated. Regular snapshots generally should not be changed once built, with the possible exception of very minor bugs that show up in testing (remember to make such fixes in the working directory too!). Sometimes it is useful to create a special snapshot to use as a development sandbox for experiments or major changes; such snapshots should use a name that does not follow the standard convention (ie., not snapshot_YYYYMMDD).

For operational uses, it generally best to use **/SCF/IDL/lib/scf_visualizer/ops/**, which should link to the latest Visualizer release snapshot; or, just execute the command **glas_visualizer**.

There are several shell scripts in scf_visualizer/working/ and each snapshot directory that allow each version to be run easily, without affecting other parts of the software:

run_visualizer_dsw.ksh – Run using the working version of data_selection (dsw = *d*ata *s*election *w*orking, etc.)
run_visualizer_dso.ksh – Run using the ops version of data_selection

These scripts are copies of the run_visualizer.ksh in /SCF/bin/working/ and /SCF/bin/ops/ respectively, *except* that they override the value of the environment variable SCF_WORKING. **They must be kept synchronized with the run_visualizer.ksh's in /SCF/bin/*/ when those change.** Different scripts are provided because the working data_selection is not always stable, whereas ops is not always up to date.

To run a snapshot Visualizer under the IDL Virtual Machine (VM), execute the script:

run_vm_visualizer.ksh

from the desired snapshot directory (or ops). This requires that there be an icesatvis_ds.sav for the snapshot (see section 17.1), and will not work for the working directory. (Although there is a prototype of run_vm_visualizer.ksh in /SCF/IDL/lib/scf_visualizer/working/, unlike the run_visualizer_ds*.ksh it *cannot* be used to run the working-version Visualizer. It is copied and edited by snapshot.mk for the specific snapshot. **It is a copy of /SCF/bin/working/run_vm_visualizer.ksh, except for the value of $SCF_VISUALIZER, and must be kept synchronized with it.**) The IDL VM is included in a full IDL installation, or can be obtained for free at http://www.rsinc.com/download/. When the IDL VM splash screen appears (note that it may come up hidden beneath other windows), click the "Click to Continue" button. From this point you will be in the Select from Specific Data Set option of the Visualizer.

Visualizer source files should have permissions rw-rw-r-- (at least). (IDL .pro files do not require x permission, although it won't hurt.) ksh files should have rwxrwxr-x. The following commands will set permissions this way:

```
chmod  ug=rw,o=r  *.pro
chmod  ug=rwx,o=rx  *.ksh
```

Section 15.1: Updating Sources for a New GSAS Version

This section enumerates the steps that must be taken when adding a new GSAS version to the Data Visualizer source code. Since the product files don't generally change record size and fields rarely move, the Visualizer will generally work (or at least not crash) when reading new files with older product structures; however, file fields will be interpreted according to the older definition, and so fields that have changed may not be displayed correctly. Unless otherwise specified, directories are relative to /SCF/IDL/lib/. The best way to include checks for specific versions in the source code is by calling the check_version routine, and its use is highly recommended.

1. In **product_readers/gla_constants.pro**, add corresponding new version and release numbers to the "releases" and "pgeversions" arrays. If any of the flags that have definitions in this file have changed, update those too.

2. In **product_readers/**, create new versions of the **glaNN_pvV_V__define.pro**, **glaNN_algvV_V_init.pro**, **glaNN_vV_V_prod2alg.pro**, and **unpack_alt_flags_vV_V.pro** (where NN is the product number and V_V is the version) files as needed. You can edit the old and new versions in the cp_version_all_pro.ksh script and run to do this. This will copy the define.pro, init.pro, and prod2alg.pro files for all products with the old version in the name to files with the new version in the name. It will also copy and change the version of the unpack_alt_flags.pro file.

3. Edit the old and new versions in the change_version_pro.ksh script and run. This will change the old versions to the new versions within the define.pro, init.pro, and prod2alg.pro files for all products. It will also change the versions in the unpack_alt_flags.pro file.

4. Modify the define.pro, init.pro, and prod2alg.pro files as necessary. Also modify the unpack_alt_flags.pro file if necessary.

5. For any new files created in step 2, make symbolic links (ln –s *target link*) in **/SCF/IDL/lib/scf_visualizer/working/**. The links should use relative paths from working/. **i.e. ln -s ../../product_readers/*v5_4* .**

6. Add the new files to the appropriate macros in **/SCF/IDL/lib/scf_visualizer/snapshot.mk**.

7. In the **product_readers/glaNN_reader__define.pro** files for products for which new files were created in step 2, add appropriate clauses to the CASE statements in the Prod2Alg and Init methods. The defaults (ELSE clauses) should usually call the new version, too.

8. In the **product_readers/glas_alg_printfield.pro** file add lines for the new version.

9. In **working/set_env_pc.pro,** update the DATA_RELEASE keyword.

10. In **working/**, modify the thumbnail generators (**series_*__define.pro, wftngen__define.pro, lidargen__define.pro, lpagen__define.pro, elevprof__define.pro,** and **cldimggen__define.pro**) for uses of products/fields that have changed in the new version, and update as needed.

Section 15.2: Creating a Visualizer Snapshot

- You must be on Icesat0 to create a snapshot, because some files are pulled from parts of /SCF that are not crossmounted to other machines. To build a snapshot, follow these steps: (You can do this from your own account, you only need to have appropriate permissions to read and write the files and directories. You do not need to be in the scf account.)

    1. cd to /SCF/IDL/lib/scf_visualizer.

    2. If any source files have been added to or removed from the Visualizer, edit the Makefile **snapshot.mk** appropriately, adding the names of the new files to the appropriate macro definitions.

    3. Type the following command:

        make –f snapshot.mk NAME=*snapshot_name*

        Watch for the line that says the snapshot was created successfully.

    4. Edit the text file *snapshot_name/*SNAPSHOT_NOTES to describe the differences from the previous snapshot.

    5. Test the snapshot by running one of the run_visualizer_ds*.ksh scripts.

  Snapshots are usually named **snapshot_YYYYMMDD**, where YYYYMMDD is the date. (This date format has the advantage of sorting correctly.) However, other names may be used for special purposes. The snapshot build process copies all the required files into the snapshot directory and its help subdirectory, chasing links as necessary. In addition, the three **run_visualizer_ds*.ksh** and the **run_vm_visualizer.ksh** files are copied and automatically edited to run the snapshot Visualizer, instead of working/.

  When a new release has been made, create symbolic links /SCF/IDL/lib/scf_visualizer/ops and /SCF/IDL/lib/scf_visualizer/RELEASE_*relnum* (where *relnum* is the versioned release number, eg. 200501.0) that point to the released snapshot directory:

  ln -s *snapshot_name* ops
  ln -s *snapshot_name* RELEASE_*relnum*

- If a snapshot has failed to build, or otherwise must be removed, type this command:

make –f snapshot.mk remove NAME=*snapshot_name*

- To create a snapshot that includes the crossovers front-end, type the command:

  make –f snapshot.mk NAME=*snapshot_name* XOVERS=1

  The crossovers version is for internal Goddard use only, it is not intended for outside distribution!

- To create the IDL save file for use with the IDL Virtual Machine (VM) and the IBM-PC version from an existing snapshot, type the command:

  make -f snapshot.mk makevm NAME=*snapshot_name*

  or equivalently (for now),

  ./prepare_save_file.tcl *snapshot_name*

  This is not currently a part of the standard build process, and must be performed as a separate step. This creates a file called **icesatvis_ds.sav** in the snapshot directory. Note that this will *not* work for the working directory, because the script adds all .pro files in the directory to the save file. The **run_vm_visualizer.ksh** script can be used to run the snapshot's VM version, provided the icesatvis_ds.sav for the snapshot has been created. (Although there is a prototype of run_vm_visualizer.ksh in the working directory, unlike the run_visualizer_ds*.ksh it can not be used in that directory. It is copied and edited by snapshot.mk for the specific snapshot.)

- snapshot.mk also has a variety of other rarely-used, special-purpose options and overridable macros that are documented in comments in that file.

Section 16: Target of Opportunity Check

The Design:
A cron job, run_too.ksh, runs once a day and checks all the station files in /SCF/ancillary_data/data/stations_data. If a file's extension is .txt and the date in the name is earlier than yesterday's date then the file will be processed. The station file contains information for all the accepted stations for the date in the name. The cron job calls the program /SCF/IDL/IDLreadGLAS/too_gla06_check.pro that calculates the minimum distance between the target of opportunity (TOO) and the nearest groundtrack for each station in the station file and saves the results in a png file for each station. The png files are copied to the SCF website and linked to their station names in the Target of Opportunity Updates table. After the station file is completely processed it is renamed with the extension .txt_chk so it will not be processed again.

The station files:
The station files are located at /SCF/ancillary_data/data/stations_data. They are named with the date and "stations.txt". Each file contains all of the accepted stations for that date and their information.

Each line contains:
Station name, date, time, latitude at the start point, latitude at the end point, longitude of the start point, and longitude of the end point. The last value is a flag for the plotting in the visualizer software and it is irrelevant in this application.

<u>Example: 2005-06-16_stations.txt</u>
MPL_Ny_Alesund,2005-06-16,13:00,79.417,78.417,12.98124,10.88476,1
Gr_030514_01,2005-06-16,17:49,80.31323,76.13455,301.48234,293.50722,1
Ant_021212_03,2005-06-16,18:32,-75.33750,-75.86970,252.96327,252.30384,1
White_Sands,2005-06-16,19:39,33.0683,32.8683,253.58531,253.55209,1

The program
Too_gla06_check.pro is the code that calculates the minimum distance between the target of opportunity and the nearest ICESAT groundtrack. It first reads the station file. For each TOO station, it reads the rev file to get the passID for that station time. It then determines the appropriate GLA06 file from the passID and lat/lon and reads it. The program calculates the shortest distance between the lat/lons in the GLA06 file and the station lat/lons and plots the results to a png file. This is done for each station in the station file. The png files are saved in /SCF/ancillary_data/data/stations_data/png_files directory. If the code successfully finishes processing all the stations in the station file than it renames the file with an extra extension "_chk".

Note: The plots don't include points of minimum distance >500 m. This is because high values of minimum distance are a usually a result of undefined latitude/longitude values in the GLA06 data.

Running the code:
too_gla06_check, too_file=<too_file>, glas_release=<glas_release>, image_dir=<image_dir>

where:
too_file is the station file that contains all the information on the TOO.
glas_release is the data release.
Image _dir is the directory to save the png files.

Section 17: GLAS-SCFWEB Operation Guide

1. Basic Concepts:

The URL of the SCF website is http://glas-scfweb.gsfc.nasa.gov.

The home directory of the site is glas-scfweb:/var/www/html. The codes for the main page are in this directory:

- The code for the home page is index.php.
- The code for the navigation bar is leftnav.htm.
- The code to switch to text only is switcher.php.
- The css subdirectory contains the css stylesheets.
- The includes subdirectory contains the header and footer bar codes.
- When needing to change the streetlight image for laser off/on, modify the following files:

  - /var/www/html/includes/scf_header.htm
  - /var/www/html/includesscf_movie_header.htm

So that

```
src="/images/icesat_sign_green_short.jpg" alt="GLAS laser is on" />

or

src="/images/icesat_sign_red_short.jpg" alt="GLAS laser is off" />
```

When wanting to make a modification, to find the correct code, on the menu bar run your mouse cursor over the buttons. The name of the routine will be shown. This is true for links and buttons on subsequent pages as well.

For example to modify a link in the list of documents:

- Placing a cursor over the “Documentation” button shows:
  http://glas-scfweb.gsfc.nasa.gov/Documentation/documentation.php
- Therefore, modify on the glas-scfweb machine, the file /var/www/html/Documentation/documentation.php.

All the cgi codes are in the /var/www/cgi-bin directory.

2. **Buttons:**

Upper row:

- ICESAT: Points to ICESat website on icesat.gsfc.nasa.gov
- WFF: Points to WFF GLAS website on glas.wff.nasa.gov
- UTCSR: Points to ICESat/GLAS website on www.csr.utexas.edu/glas/
- NSIDC: Points to ICESat/GLAS website on nsidc.org/daac/icesat/
- ATMOSPHERE: Points to GLAS website on glo.gsfc.nasa.gov (portions password protected)
- ISF: Points to Instrument Support Facility website on glasisf.wff.nasa.gov (password protected)

Navigation bar:

- "Home": Points to /var/www/html/index.php
- "Browse Products": Points to /var/www/html/Browse_products/browser_page.html
- "Bulletin Board": Points to /var/www/html/Bulletin_board/bulletinboard_gate.html
- "Data Tables": Points to /var/www/html/Data_tables/data_tables.php
- "Date Conversion Tool": Points to /var/www/html/Date_conversion/date_conversion.php
- "Documentation": Points to /var/www/html/Doumentation/documentation.php
- "Energy/Waveform Analysis Plots": Points to /var/www/html/Energy_analysis/energy_analysis.php
- "Instrument Updates": Points to /var/www/html/Instrument_updates/instrument_updates.php
- "I-SIPS Status Display": Points to /var/www/html/ISIPS_Status/ISIPS_Status.php
- "Presentations": Points to /var/www/html/Presentations/presentations.php
- "Product Formats": Points to Product Formats Database on glas.wff.nasa.gov
- "Product QA": Points to /var/www/html/password/product_qa_login.php
- "Product Trends Plots": Points to /var/www/html/Product_trends/product_trends.php
- "Statistics": Points to /var/www/html/Statistics/statistics.php
- "Subscriptions": Points to /var/www/html/Subscriptions/menu_page.html

3. **Directory Structure:**

The following directories are under the home directory, /var/www/html:

- Browse_products
- browser
  - help_<release>
  - <years>
    - <months>
      - <days>
        - <products>
          - <releases>
- Bulletin_board
  - buttons
- css
- Data_tables
  - tables
- Date_conversion
- Documentation
  - docs
- eboard30
  - criteria1 (product)
    - messages
  - criteria2 (misc)
    - messages
  - criteria3 (visualizer)
    - messages
  - criteria4 (data request)
    - messages
  - criteria5 (sys admin)
    - messages
  - criteria6 (instruments)
    - messages
  - criteria7 (web)
    - messages
- Energy_analysis
  - Bae
    - plots
  - ea
    - plots

    - wf
        - plots
- images
- includes
- Instrument_updates
    - tables
        - too_plots
- ISIPS_Status
    - data
        - <years>
    - images
    - includes
- movies
- password
- Presentations
    - dirs
        - <versions>
- Product_trends
    - plots
- Queen
    - images
- reference
- revFile
- Statistics
    - src
    - stat_<year>
        - html
        - images
- Subscriptions
    - gifs
    - html
    - images
    - js

Directory Descriptions:

- The Browse_products directory contains the files that access the browse products in the in /var/www/html/browser directory.
  /var/www/html/Browse_products/browser_page.html is the main script.
  Accessed by the “Browse Products” button.

- The browser directory contains the browse products in the following subdirectory structure.
  /var/www/html/Browse_products/browser_page.html is the main script that points to the files in this directory.
  Accessed by the “Browse Products” button.

  - The help_<release> directory contains the help file for each GLAS product. Each release has its own help directory.
  - The <years> directory stands for the 2003, 2004, 2005, 2006, etc. directories.
  - The <months> directory stands for the directories of all the months in the year that contain GLAS data.
  - The <days> directory stands for the directories of all the days in the month that contain GLAS data.
  - The <products> directory stands for the directories of all the products on the day that have GLAS data.
  - The <releases> directory stands for the directories of all the releases of the product that have GLAS data.

  The browser directory also has the script mv_web_files_to_dirs.pl. This script moves the browse images in the current directory to their appropriate browse directories if this needs to be done manually.

- The Bulletin_board directory contains the files that access the bulletin board messages in the in /var/www/html/eboard30 directory.
  /var/www/html/Bulletin_board/bulletinboard_gate.html is the main script.
  Accessed by the “Bulletin Board” button.

- The css directory contains the css stylesheets. Style.css is used for the main pages and plot_style.css is used for the plot pages.

- The Data_tables directory contains the I-SIPS release table, the SCF release table, and the Borg spreadsheets in Excel and HTML format. The tables are under the “tables” subdirectory.
  /var/www/html/Data_tables/data_tables.php is the main script that points to the files in this directory.
  Accessed by the “Data Tables” button.

- The Date_conversion directory contains files that access the date conversion tool.
  /var/www/html/Date_conversion/date_conversion.php is the main script.
  Accessed by the “Date Conversion Tool” button.

Note: Many of the codes for this tool are cgi codes in the /var/www/cgi-bin directory.

- The Documentation directory contains the SCF documentation in pdf format. The files are under the “docs” subdirectory. /var/www/html/Documentation/documentation.php is the main script that points to the files in this directory. Accessed by the “Documentation” button.

- The eboard30 directory pertains to the Bulletin Board and contains subdirectories for each board. /var/www/html/Bulletin_board/bulletinboard_gate.html is the main script that points to the files in this directory. Accessed by the “Bulletin Board” button.

  Each “criteria” subdirectory pertains to a specific category and contains the following files:
  - mlist.txt – each line contains the message index, message title, and the person who posted the message
  - alist.txt – the same format as the mlist.txt file; these data are for archived messages
  - slist.txt - contains the email addresses of the subscribers

  The categories are as follows:
  - criteria1: Product
  - criteria2: Misc
  - criteria3: Visualizer
  - criteria4: Data Request
  - criteria5: Sys Admin
  - criteria6: Instruments
  - criteria7: Web

  The “messages” subdirectories contain the Bulletin Board messages for each specific category. Each message has an index associated with it.

- The Energy_analysis directory contains the energy and waveform analysis plots for each laser campaign. The energy analysis plots are in “ea/plots” and the waveform analysis plots are in “wf/plots”. Sungkoo Bae’s time series plots are in “Bae/plots”. /var/www/html/Energy_analysis/energy_analysis.php is the main script that points to the files in this directory. Accessed by the “Energy/Waveform Analysis Plots” button.

- The images directory contains images (jpg, gif, etc.) that are used as buttons, pictures, etc.

- The includes directory contains codes to include the NASA header and footer, pop-up windows, movies, etc. in the website.

- The Instrument_updates directory contains the Instrument Updates Table, Instrument Temperatures Table, Target of Opportunity Updates Table, Pointing Updates Table, and Real Time and Saved Command Updates Table. The tables are in the “tables” subdirectory. The “tables/too_plots” subdirectory contains the distance plots for each TOO location in png format.
  /var/www/html/Instrument_updates/instrument_updates.php is the main script that points to the files in this directory.
  Accessed by the “Instrument Updates” button.

- The ISIPS_Status directory contains the files for the I-SIPS Processing Status Display pages.
  /var/www/html/ISIPS_Status /ISIPS_Status.php is the main script that points to the files in this directory.
  Accessed by the “I-SIPS Status Display” button.

  - The data subdirectory contains several <years> subdirectories (i.e. 2003, 2004, 2005, 2006, etc.) Each year subdirectory contains csv files.
  - The images subdirectory contains png files.
  - The includes subdirectory contains php files.

- The movies directory contains .swf movie files.

- The password directory contains the password-protected login page for the Product QA.
  /var/www/html/password/product_qa_login.php is the main script that points to the files in this directory.
  Accessed by the “Product QA” button.

- The Presentations directory contains Science Team presentations in ppt or pdf format under versioned subdirectories corresponding to the date of each GLAS Science Team Meeting.
  /var/www/html/ Presentations/presentations.php is the main script that points to the files in this directory.
  Accessed by the “Presentations” button.

- The Product_trends directory contains Product Size Plots for each laser campaign. The plots are in the “plots” subdirectory.
  /var/www/html/Product_trends/product_trends.php is the main script that points to the files in this directory.
  Accessed by the “Product Trends Plots” button.

- The Queen directory contains pictures of the Queen Elizabeth II’s visit to GSFC in 2007.
  /var/www/html/Queen/queen_visit.php is the main script that points to the files in this directory.
  Accessed under the “NASA/GSFC Fact” on the home page.

- The reference directory contains the original NASA website templates for reference.

- The revFile directory contains the rev table used by the Date Conversion Tool.

- The Statistics directory contains files for the distribution and yearly summaries.
  /var/www/html/Statistics/statistics.php is the main script that points to the files in this directory.
  Accessed by the “Statistics” button.

  - The src subdirectory contains perl scripts to create the summary calendar pages.
  - The stat_<years> subdirectories (i.e. stat_2003, etc.) contain html and images subdirectories. The html subdirectory contains html files for each date and laser_type txt files. The images subdirectory contains plots of subscriptions and special requests for each date in png format.

- The Subscriptions directory contains files for the subscription pages.
  /var/www/html/Subscriptions/menu_page.html is the main script that points to the files in this directory.
  Accessed by the “Subscriptions” button.

  - The gifs subdirectory contains gif images used in the subscription menu.
  - The html subdirectory contains html files for each subscription.
  - The images subdirectory contains jpeg files for each subscription.
  - The js subdirectory contains javascript files used for the subscription menu.

4. How to update the document list

Update the document list whenever you have a new version of a document to post on the web or if you want to add a new document to the list.

<u>To post a new version of a document:</u>
First, convert the file from document format to PDF format. To do so, you must have the software on your PC that can do it. (i.e. Adobe Acrobat software). If the document needs the title page already in PDF format, click on “Document - Insert Pages”, select the correct title page and insert the page at the beginning of the document. On the left side there is a text option that will allow you to edit the title page. Update the version and date and save the entire document. After the file is in PDF format, copy it to glas-scfweb:/var/www/html/Documentation/docs. Copy the original Word document to the subdirectory, “Word_docs” (for safe-keeping). Lastly, update the version in the /var/www/html/Documentation/documentation.php file.

5. How to update the I-SIPS Release table

- Receive Excel spreadsheet created by Suneel.
- In Excel, save as web page called GLAS_Product_Releases.htm.
- Transfer htm file to glas-scfweb:/var/www/html/Data_tables/tables.

6. Configuration of glas-scfweb

Since glas-scfweb is a webserver, its configuration is different than that of the other remote sites.

SCF directory Structure

The following subdirectories are under glas-scfweb:/SCF:

bin/ops
dist
ingest
src/perl
tmp
utils

The /SCF directory contains software necessary to move files transferred from the mSCF and to monitor the system. Web files are kept under /var/www/html. Perl scripts are kept under /var/www/cgi-bin.

Perl Configuration

Note: Link /usr/bin/perl to /opt/perl/bin/perl

Perl modules needed:

GD::Text
GD::Graph
DBD::mysql

DBI

Linux software needed by above:

libgd
libgd-devel
libjpeg
libpng
libfreetype
mysql
mysql-devel

Tcl Path

Tcl scripts need the following first line:

#!/usr/bin/tclsh

Ssh Path

The full path for ssh is the following:

/usr/bin/ssh

Datecon2005

Source code for datecon2005 is in /SCF/src/datecon2005. The executable must be copied to /usr/bin to be utilized by the website software.

Crontab

The following entries are run under the scf crontab:

```
* * * * * /SCF/src/perl/ops/run_scripts.ksh
00 * * * * /SCF/bin/ops/run_check_dist_rscf.ksh >> /SCF/tmp/cron_dist_errors.txt 2>&1
00 05 * * * /var/www/html/subscriptions/src/run_daily_stats_perl.ksh  >> /SCF/tmp/cron_daily_stats_errors.txt 2>&1
```

Section 18: Running Energy/Waveform Analysis

Steps:

1. Go to the **/SCF/IDL/EnergyAnalysis/working** directory on icesat0

2. Delete previously created Energy and Waveform Analysis output text, png and html files:

   ```
   $ rm energy_analysis_L2B.txt
   $ rm wfdailystats_L2B.txt
   $ rm energy_analysis_L2B*.html
   $ rm wf_L2B_analysis*.html
   $ rm energy_analysis_L2B.png
   $ rm wfstats_L2B*.png
   ```

   Note: in this case L2B was previously run.

3. Edit control file **scfplots_ctrl.dat**. Below is an example of the control file:

   ```
   =SCFPLOTS
   Version = 5.2
   Input_File=/SCF/product_sets/current/L2B/GLA01_026_2107_001_1284_1_01_0001.P0372
   Input_File=/SCF/product_sets/current/L2B/GLA01_026_2107_001_1284_2_01_0001.P0372
   .....
   DoWFAnalysis=TRUE
   MakeWFPlots=TRUE
   LogFile=wf_log.txt
   Wfstats_Plot_File=wfstats_L2B.ps
   WfGranuleStats_File=wfstats_L2B.dat
   WFDailyStats_File=wfdailystats_L2B.txt
   DoEnergyAnalysis=TRUE
   MakeEnergyPlot=TRUE
   Energy_Title=Laser 2b 91-day
   Energy_OutFile=energy_analysis_L2B.txt
   Energy_Plot_File=energy_analysis_L2B.ps
   ```

GainCutoff = 30 50 80 30
GainCutoffUTC = 138110400 138974400 140529600 150000000
=END

3a. Include the GLA01 input files for which the Energy and Waveform analysis are being generated. For laser L2B, Release 26 for example, a list can be generated like this:

$ ls /SCF/product_sets/current/L2B/GLA01_026* > l2b.lis

Include L2B.lis in the control file. Don't forget to include the "Input_File=" keyword for each file.

NOTE: If there are more than 1000 GLA01 files to process, include only 1000 or so. Repeat the whole process as many times as necessary for the remaining of the GLA01 data files. The results will be appended to energy_analysis_L2B.txt. Therefore, do not delete any output files until all the GLA01 files for a particular laser have been processed. Also, don't worry about including an input file more than once. Duplicate results will be eliminated.

3b. Modify the file names to reflect the laser campaign being processed, in this example L2B:

Wfstats_Plot_File=wfstats_**L2B**.ps
WfGranuleStats_File=wfstats_**L2B**.dat
WFDailyStats_File=wfdailystats_L2B.txt
Energy_Title=Laser **2b 91-day**
Energy_OutFile=energy_analysis_**L2B**.txt
Energy_Plot_File=energy_analysis_**L2B**.ps

3c. Modify the flags to indicate what processing you want done:

- DoWFAnalysis=TRUE Perform waveform analysis on GLA01 files
- MakeWFPlots=TRUE Create waveform analysis plots
- DoEnergyAnalysis=TRUE Perform energy analysis on GLA01 files
- MakeEnergyPlot=TRUE Create energy analysis plots

If you don't want a particular processing done, set the keyword to FALSE.

Note: The analysis routines create the text and data files. The plot routines read these files to create the postscript files. Therefore, if for example, you forgot to change the Energy_Title for the new processing, after you change it, set

MakeEnergyPlot=TRUE and the other flags to false. This will only regenerate the energy analysis plots from the Energy_OutFile.

4. Run **scfplots.pro.** To run:

   $idl
   IDL> **scfplots, 'scfplots_ctrl.dat'**

   The output from this program should look like this:

   energy_analysis_L2B.txt
   energy_analysis_L2B.txt.bak
   energy_analysis_L2B.ps
   wfdailystats_L2B.txt
   wfstats_L2B_1.ps
   wfstats_L2B_2.ps
   wfstats_L2B_3.ps
   wfstats_L2B_4.ps
   wfstats_L2B_5.ps
   wf_log.txt

   Note: Check the log file wf_log.txt to make sure program completed without problems. If the program ends with an error opening a GLA01 file, it is likely that too many GLA01 files were included in the control file and the IDL program run out of memory. In that case, the number of GLA01 data files needs to be reduced and the process rerun.

5. Convert postscript ps files to png file format so that they can be viewed on the glas-scfweb web site.

   $ **ps_to_png energy_analysis_L2B.ps energy_analysis_L2B.png**
   $ **ps_to_png wfstats_L2B_1.ps wfstats_L2B_1.png**
   $ **ps_to_png wfstats_L2B_2.ps wfstats_L2B_2.png**

   Repeat this for every postscript file created in the previous step.

   Note: The ps_to_png.ksh script will automatically delete the postscript file after being invoked, even if no png ouput file name is given. If the postscript file energy_analysis_L2B.ps is mistakenly deleted, it can be recreated quickly, without having to run

the whole process again. Just edit the control file to create only the energy analysis plots again as per section 3c. Likewise, the wfstats_L2B_1.ps, etc. postscript files can be regenerated as well, if need be.

6. Convert output text files (output from step 4) to html files so that they can be viewed on the glas-scfweb web site:

   $ **idl** (Bring up IDL)
   IDL> **energy_analysis2html,'energy_analysis_L2B.txt','energy_analysis_L2B_table.html'**

   The output from this IDL program should look like this:

   energy_analysis_L2B_table.html

7. Do the same for waveform analysis:

   $ **idl** (Bring up IDL)
   IDL> **wf_analysis2html,'wfdailystats_L2B.txt','wf_L2B_analysis'**

   The output from program should look like this:

   wf_L2B_analysis_SEAICE.html
   wf_L2B_analysis_OCEAN.html
   wf_L2B_analysis_LAND.html
   wf_L2B_analysis_ICESHEET.html
   wf_L2B_analysis_COMBINED.html

8. Copy all png and html files just generated to **glas-scfweb:/var/www/html/inst_update**

   $ **scp *L2B*.png glas_scfweb:/var/www/html/inst_update/**
   $ **scp *L2B*.html glas_scfweb:/var/www/html/inst_update/**

9. Log onto glas_scfweb and change the Release (if necessary) in the html file:

   glas-scfweb:/var/www/html/inst_update/wf_ L2B_analysis.html

10. Test '**Energy/Waveform Analysis**' (glas-scfweb.gsfc.nasa.gov) to make sure everything is working as expected.

Section 19: Creating GLAS Pointing Mode Plots

These are step-by-step instructions for creating PNG plots of GLAS ground tracks across Greenland and Antarctica, color-coded by pointing mode for display on the SCF website. **All files are on icesat0**.

1) Determine which laser period needs plots created

The order of processing goes like this: RTSCM_POINTING database table or pointing file from Peggy -> track file -> ps file -> png file. The ps and png files are by laser period but the track files and point tables are by time and therefore may contain several laser periods. To determine which laser period(s) need ps files created:

- cd /SCF/product_sets/jguerber/TOO_Tracks/Antarctica
- ls *.ps
- cd /SCF/product_sets/jguerber/TOO_Tracks/Greenland
- ls *.ps

To see the ps files in the Antarctica and Greenland directories. Any ones missing need to be created.

2) Create track file for each laser period

The file **/SCF/product_sets/jguerber/TOO_Tracks/PointingTables/README.txt** lists the track file corresponding to each laser period. If a laser period is not listed, then a track file needs to be created for it. Usually it can be created from the RTSCM_POINTING database table or an archived version of it. If the laser period is from before we started creating the database table then the track file must be created from a pointing update file supplied by Peggy Jester.

2a) Create track file using database table

Kristine or Jairo can use the Navicat export wizard to produce ASCII tables from the RTSCM_POINTING database table; they should include all columns, and be tab-separated. The final line will not end in a newline, which must be added manually with their favorite text editor.

2b) Create track file using Peggy's file

These file must be reformatted to match those produced by Navicat, and the pointing flags determined. To do so, use the program /SCF/IDL/TOO_Tracks/pointing_cvt_peggy.pro:

- cd /SCF/IDL/TOO_Tracks

- idl

IDL> !path = '/SCF/IDL/TOO_Tracks/:' + !path

IDL> .run pointing_cvt_peggy.pro

Note: You only need to run these two commands once during the IDL session

IDL> pointing_cvt_peggy, 'infile', 'outfile'

where infile and outfile are the names of the input and output (reformatted) files respectively, enclosed in quotes. See the /SCF/product_sets/jguerber/TOO_Tracks/PointingTables/README.txt file for available file names.

3) Create ps files from the track files

Use the program **/SCF/IDL/TOO_Tracks/tootracks.pro** to create ps files from the track files created above:

- cd /SCF/IDL/TOO_Tracks

- idl

IDL> .run tootracks.pro

Note: You only need to run this command once during the IDL session

IDL> **tootracks, 'path/trackfile', 'Lxx', /grn, outps='path/psfile.ps', /nonadir, /noref**

where:

'path/trackfile': name of the track file containing the laser period you are creating the ps file for. The name must include the path and it must be in quotes.

Note: Path for track files is: '/SCF/product_sets/jguerber/TOO_Tracks/PointingTables/'

'Lxx': string specifying the laser period, such as 'L2C', 'L3G', etc. Must be in quotes.

/grn or /ant: keyword indicating whether to plot Greenland (/grn) or Antarctica (/ant).

'path/psfile.ps': name of the output PostScript file. The name must include the path and it must be in quotes. It is recommended that it include the laser campaign and either Ant or Grn as appropriate. i.e. Grn_L3K_pointing.ps or Ant_L3K_pointing.ps

Note: Path for Greenland files is: '/SCF/product_sets/jguerber/TOO_Tracks/Greenland'
Path for Antarctica files is: '/SCF/product_sets/jguerber/TOO_Tracks/Antarctica'

/nonadir, /noref: keywords indicating to NOT plot the nadir or reference tracks. Other options are /notoo (no TOO tracks) and /noscan (no scan tracks). /nonadir, /noref is the usual combination.

This program is further documented in /SCF/IDL/TOO_Tracks/README.tootracks.

Examples:

IDL> tootracks, '/SCF/product_sets/jguerber/TOO_Tracks/PointingTables/RTSCM_POINTING_2009_FEB_25.txt', 'L3K', /grn, outps='/SCF/product_sets/jguerber/TOO_Tracks/Greenland/Grn_L3K_pointing.ps', /nonadir, /noref

IDL> tootracks, '/SCF/product_sets/jguerber/TOO_Tracks/PointingTables/RTSCM_POINTING_2009_FEB_25.txt', 'L3K', /ant, outps='/SCF/product_sets/jguerber/TOO_Tracks/Antarctica/Ant_L3K_pointing.ps', /nonadir, /noref

3a) Check that the ps files have been created and look ok

To display the image of the newly created Greenland file:

- display /SCF/product_sets/jguerber/TOO_Tracks/Greenland/<psfile> where <psfile> is the new ps file

To display the image of the newly created Antarctica file:

- display /SCF/product_sets/jguerber/TOO_Tracks/Antarctica/<psfile> where <psfile> is the new ps file

4) Create png files from ps files

- Scp the ps files created above to **glas-scfweb:/var/www/html/Instrument_updates/point_mode/plots.**

- Convert the ps files to png files by typing: **tootracks_ps2png2.ksh <psfile.ps>**

where psfile.ps is the PostScript file you want converted. The output file will have the same base name with a .png extension. (A different output name can be supplied as an optional second argument.)