NASA/TM—2005–209997/VER201103.0/VOL. 7

ICESat (GLAS) Science Computing Facility Document Series

Volume 7.0
SCF Operator's Guide
Version 201103.0

Kristine Barbieri
Anita Brenner

National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

April 2011

TABLE OF CONTENTS

TABLE OF CONTENTS III

SECTION 1: INTRODUCTION 1

SECTION 2: RELATED DOCUMENTATION 2

SECTION 3: FUNCTIONS OF THE MAIN SCF 3

SECTION 4: RESPONSIBILITIES OF THE MAIN SCF OPERATOR 4

SECTION 5: SITE NAMES 5

SECTION 6: DATA MOVEMENT 6

SECTION 7: DATA 8

SECTION 7.1: REV FILE 9

SECTION 8: MYSQL DATABASE TABLES 10

SECTION 8.1: MYSQL DATABASE COMMANDS 12

SECTION 9: HANDLING DATA REQUESTS 14

SECTION 9.1: SUBMITTING REQUESTS TO MSCF 15

SECTION 9.2: SUBMITTING QA UPDATES 18

SECTION 9.3: MODIFYING AND CANCELLING SUBSCRIPTIONS 20

SECTION 10: MONITORING DATA AND REQUESTS 22

SECTION 11: AUTOMATED PROCESSING 25

SECTION 11.1: EMAIL-TRIGGERED PROCESSES 26

SECTION 11.2: CRON PROCESSES 27

SECTION 11.3: PROCESS MESSAGES 30

SECTION 12: TROUBLESHOOTING GUIDE 36

SECTION 12.1: DATA TRANSFER PROBLEMS 37

SECTION 12.2: DISK FULL PROBLEMS 39

SECTION 12.3: FORK/ENTROPY/PRNG PROBLEMS 42

SECTION 12.4: MMAP ERRORS 44

SECTION 12.5: WHEN A SPECIAL REQUEST IS NOT FULFILLED 45

SECTION 12.6: WHEN DATA PROCESSING STOPS 47

SECTION 12.7: WHEN SUBSCRIPTION PROCESSING STOPS 48

SECTION 12.8: SENDING A PAN TO THE I-SIPS 50

APPENDIX A: A DAY IN THE LIFE CHEAT SHEET 51

APPENDIX B: ABBREVIATIONS & ACRONYMS 52

SECTION 1: INTRODUCTION

The Geoscience Laser Altimeter System (GLAS) is the sole instrument aboard the Ice, Cloud, and land Elevation Satellite (ICESat). Its mission is to use a lidar alimeter to measure surface elevations, especially of the polar ice caps. As the lidar beam travels through the atmosphere it reflects back from cloud and aerosol layers and therefore gives atmospheric data as well.

Data from the GLAS instrument is processed by the ICESat Science Investigator-led Processing System (I-SIPS). The processed data is then sent to the main Science Computing Facility (mSCF) for dissemination to the GLAS science team. The science team members reside at various remote Science Computing Facilities (rSCF's) across the country. To receive GLAS data they have to submit subscriptions to the mSCF. If they need data that they don't ordinarily subscribe to, they can submit a special request for the data. The processing of data received from the I-SIPS and the fulfillment of subscriptions and requests was designed to be as automated as possible. However, an SCF operator is necessary to monitor the data processing, submit and monitor request fulfillment and troubleshoot problems as necessary.

SECTION 2: RELATED DOCUMENTATION

- *SCF Architectural Design Document*
- *SCF Data Request Software Detailed Design Document*
- *SCF Data Request Software User's Guide*
- *SCF Data Visualization Software User's Guide*
- *SCF Operator's Guide*
- *SCF Interface Control Document*
- *SCF Interface Software Update Document*
- *SCF Interface Software Installation Guide*
- *SCF Interface Software Detailed Design Document*
- *SCF Interface Software Operator's Guide for rSCF*
- *SCF Interface Software Operator's Guide for mSCF*
- *I-SIPS Interface Software Operator's Guide for ISF*
- *I-SIPS Interface Software Operator's Guide for UTCSR*
- *Interface Control Document Between I-SIPS/ISF and CSR*

SECTION 3: FUNCTIONS OF THE MAIN SCF

The primary function of the mSCF is to disseminate GLAS data processed by the I-SIPS to the GLAS science team. This is accomplished by fulfilling subscriptions and special requests for data. Ancillary to this basic function are other functions:

- Store all GLAS level 2 and selected GLAS level 1 products with a data management system
- Maintain request and distribution history
- Develop and maintain GLAS analysis and visualization tools
- Provide the GLAS science team with product QA
- Communicate product QA updates from the science team to the I-SIPS
- Maintain a bulletin board for science team comments on and discussion of GLAS software, products, and issues.
- Act as the remote SCF for GSFC-based science team members
- Create or receive level 3 and 4 products and distribute them to the I-SIPS

SECTION 4: RESPONSIBILITIES OF THE MAIN SCF OPERATOR

The SCF operator is necessary to monitor the data processing and request fulfillment and troubleshoot problems as necessary. These tasks are accomplished by:

- Understanding the data flow from I-SIPS to msCF to rSCF's
- Understanding and accessing the MYSQL database tables
- Understanding how the data transfers work and knowing what to look for if they fail
- Knowing and using basic troubleshooting techniques
- Submitting subscriptions and data requests from the science team
- Understanding how the data request software works and knowing what to look for if it fails
- Knowing what to do when error messages are received
- Backing-up the glas-scfweb server once a week.

SECTION 5: SITE NAMES

Acronyms have been chosen to represent the various GLAS sites. These site names will be used in the Product Delivery Record and Product Acceptance Notice file names, the input and output cache names on the I-SIPS and mSCF, and in documentation. The table below gives the site name for each institution:

Institution	Site Name
GSFC ICESat Science Investigator-led Processing System	I-SIPS
GSFC Main Science Computing Facility	mSCF
GSFC Instrument Support Facility	ISF
GSFC Altimetry	ALT
University at Buffalo	BUF
GSFC Lidar	LIDAR
Colorado State University	LRSCF
Massachusetts Institute of Technology	MIT
University of Colorado, National Snow and Ice Data Center	NSIDC
Ohio State University	OSU
GSFC Hydrology	SIMPL
University of California, San Diego, Scripps Institution of Oceanography	UCSD
University of Texas, Austin, Center for Space Research	UTCSR
Wallops Flight Facility	WFF
Remote Science Computing Facility (ALT, BUF, etc. collectively or generally)	rSCF
SCF web server (glas-scfweb.gsfc.nasa.gov)	SCFWEB

SECTION 6: DATA MOVEMENT

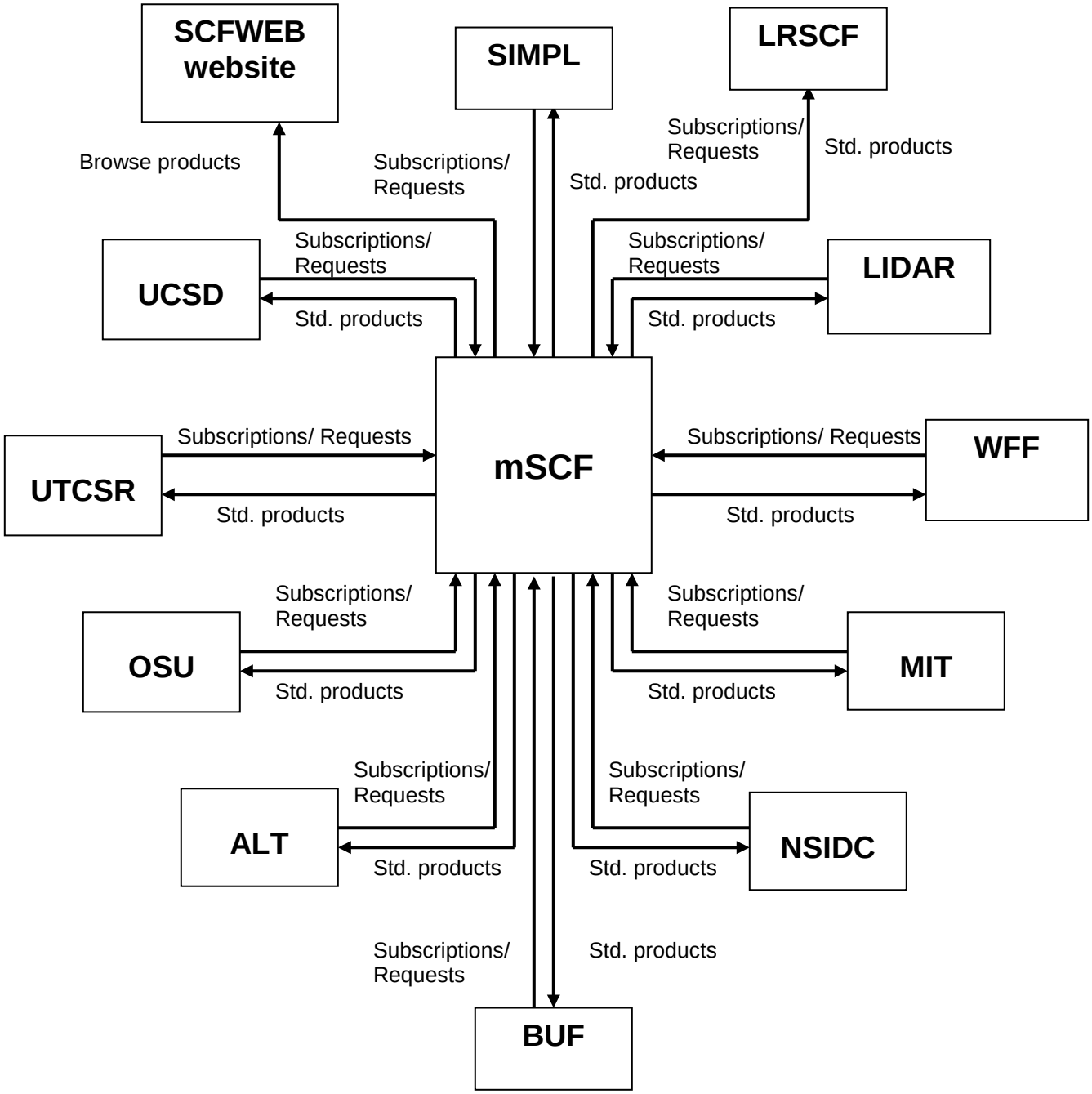

Figure 6-1: Data flow between the mSCF and the rSCFs

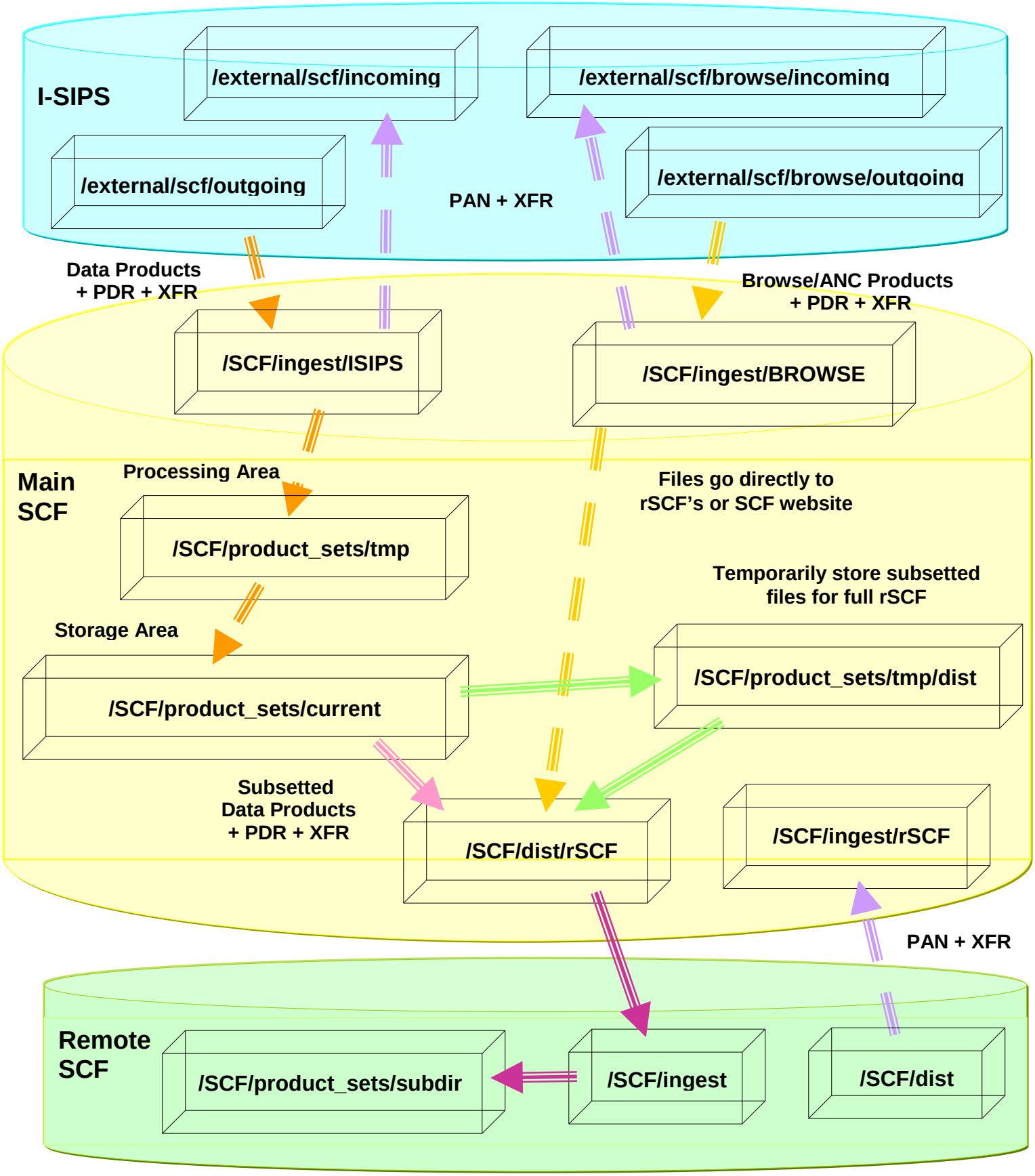

Figure 6-2: Data flow between the I-SIPS, mSCF and an rSCF

SECTION 7: DATA

Data created by the I-SIPS and transferred to the mSCF on a subscription basis includes:

- All granules of all products GLA01 - GLA15
- Browse products from all products, GLA01 – GLA15 (HDF files)
- Updated rev file (ANC28 file) as necessary
- Updated reference orbit track files as necessary

Data kept on the mSCF includes:

- All granules of GLA01, GLA05-GLA15
- LPA granules of GLA04
- Browse products from all products, GLA01 – GLA15 (HDF files) are put on scfweb
- Rev file (ANC28 file)
- Reference orbit track files

Data created on the mSCF includes:

- One pass table for each product file
- One unique record index table for each product file
- One bin table for each product file
- A georeference table for each bin table

Data created by the mSCF and transferred to the rSCF includes:

- Any portion of products GLA01-GLA15 as requested via subscription or special request
- One pass table for each sub-setted product file
- One unique record index table for each sub-setted product file
- One bin table for each sub-setted product file
- A georeference table for each bin table
- Browse products for each sub-setted product file (png files - if requested)
- Updated rev file (ANC28 file) as necessary

SECTION 7.1: REV FILE

The rev file is a binary file that contains the start times for each rev (track, pass) and is located in /SCF/ancillary_data/rev_files. It is created at the I-SIPS every time they receive a predicted orbit file from UTCSR. It is a cumulative file, therefore the new file may replace the old file. It is downloaded from the I-SIPS as an ANC28 file and moved to the file name, total_rev_file.dat, by the script, /SCF/src/perl/ops/update_rev_file.pl. This script saves the old rev file as total_rev_file.dat.sav and also creates an ASCII version of the file called total_rev_file.txt. Since the remote sites need the rev file as well, this script also transfers the binary and text versions of the rev file to each rSCF. The rev file should be received at the mSCF before data are received or they will not be able to be processed by the data processing code.

If data have been in the /SCF/product_sets/tmp directory for a while, along with a PDR and XFR, it is likely that the rev file needs to be updated. Try invoking /SCF/bin/ops/test_process_data.ksh to see if this is the case. If so, the program will end rather rapidly and will print out the last rev times of the data and the file. If the last rev time of the data is larger than the last rev time in the file, then a new rev file must be obtained from the I-SIPS. Inform the I-SIPS operators by sending an email to ops@isipspr1.nascom.nasa.gov or by submitting a Mantis problem report indicating that the SCF needs an updated rev (ANC28) file.

If an ANC28 file is in the /SCF/ancillary_data/rev_files directory, but for some reason has not been processed, copy the FN and XFR from the spare_FN_file subdirectory and replace the ANC28.DAT file name with the real ANC28 file that needs to be processed. The next time that /SCF/bin/ops/run_scripts_mscf.ksh is run, the ANC28 file should be processed.

SECTION 8: MYSQL DATABASE TABLES

The MYSQL database maintains data request and data distribution histories. Errors are also tracked on it. Instrument status information is used for determining if data are good enough to be distributed. This information is also posted on the SCF website along with the target of opportunity information.

The following tables contain rSCF - mSCF subscription information:
- USER
- SUBSCRIPTION_USER
- REQUEST_PRODUCT_SEG
- REQUEST_TRACKS
- REQUEST_CYCLES

The following tables contain rSCF - mSCF subscription distribution information:
- DISTRIBUTION
- DISTRIBUTION_FILES
- SUBSCRIPTION_INPUT_FILES

The following tables contain rSCF - mSCF special request information:
- USER
- SPECIAL_REQUEST_USER
- REQUEST_PRODUCT_SEG
- REQUEST_TRACKS
- REQUEST_CYCLES

The following tables contain rSCF - mSCF special request distribution information:
- DISTRIBUTION
- DISTRIBUTION_FILES
- SPECIAL_REQUEST_PID
- [rSCF]_PRODUCT_ID

The following tables contain mSCF - I-SIPS subscription information:
- ISIPS_SUBSCRIPTIONS

The following tables contain mSCF-I-SIPS subscription distribution information:
- CREATION
- ISIPS_PRODUCT_ID

The following tables contain mSCF - I-SIPS special request information:
- ISIPS_REQUESTS

The following tables contain mSCF - I-SIPS special request distribution information:
- ISIPS_DISTRIBUTION

The following table contains updated product QA information:
- QA_PRODUCT_UPDATE

The following table contains target of opportunity information:

- TOO_UPDATE

The following tables contain instrument status information:
- INSTRUMENT_UPDATE
- RTSCM_UPDATE

SECTION 8.1: MYSQL DATABASE COMMANDS

To access the database tables use the following MYSQL commands:

- To get into MYSQL: mysql --user=scf -p (type password at prompt)
- To use a database: use *database_name* *(i.e. use ops)*
- To show list of databases: show databases;
- To show list of tables: show tables;
- To repair a table: checkmysql –r *database_name* *table_name*
- To repair all tables in a database: checkmysql –r *database_name*
- To create a table: create table *new_table* (
  RequestId varchar(5) not null,
  Release float not null,
  Segment int not null,
  Description longtext,
  Date datetime not null);

The easiest way to add a column to a table is to drop the table and recreate it (unless you cannot lose the data). For convenience, table creation commands are in /SCF/tables. Please update the appropriate file if the format of a table changes.

- To remove a table: drop table *table*;
- To see a description of a table: desc *table; (i.e. desc USER;)*
- To see everything in a table: select * from *table*;
- To see everything in a table for a specific criterion: select * from *table* where *column=value;* (i.e. *userName="kris"*;). Use quotes around strings.
- To see everything in a table for a non-specific criterion:
  select * from *table* where *column* like *value; (i.e. userName like "k%%%"*;). % is a 1 character wild card.
- To see a specific column in a table: select *column* from *table;*
  (i.e. select *institute from USER;)*
- To see specific columns in a table for specific criteria: select *column1, column2* from *table* where *column3=value and column4=value; (i.e.* select *institute, userId from USER where userName="kris" and email="kris@nasa.gov";)*
- To update a specific column for a specific criterion:
  update *table* set *column=new_value where column=value;*

(i.e. update *USER set userName="kristine" where userName="kris";)*

- To delete rows for a specific criterion: delete from *table where column=value;*
  (i.e. delete from *USER where userName="kris";)*

- To insert a row into a table: insert into *table* values (*"Kris", 20, "NULL", now()*);
  NULL is the null value and now() gives the current time string i.e. 2002-09-26 11:45:50

- To delete all rows from a table: delete from *table;*

- To add a column to a table: alter table *table* add *column_name variable_type;*
  (i.e. alter table *USER* add *status char(1);)*

- To exit MYSQL: exit

SECTION 9: HANDLING DATA REQUESTS

SECTION 9.1: SUBMITTING REQUESTS TO MSCF

Data request emails now go to scf_manager@icesat0.gsfc.nasa.gov rather than going directly to scf@icesat0.gsfc.nasa.gov. This is so mSCF personnel can have more knowledge about what is being requested and thus more control over its outcome. Below is an example of the request email:

```
Special Request Flag =        1
User = ben
Email = ben@ess.washington.edu
Institute = UW
Product_ID = GLA01
Product_ID = GLA05
Data_Release =        0
Start_Date = 2003022000
End_Date = 2009010100
Minimum_Latitude =       60.000000
Maximum_Latitude =        84.000000
Minimum_Longitude =       -67.00000
Maximum_Longitude =       -20.00000
8_Day_Repeat = 0
8_Day_Tracks =  31
8_Day_Tracks =  40
8_Day_Tracks =  41
91_Day_Repeat = -1
91_Day_Tracks = -1
Output Directory = /SCF/product_sets/V10_test_data
QL_flag = 0
```

Below is a breakdown of the request email components:

- Special Request Flag: 0 indicates subscription, 1 indicates special request
- User = user's name
- Email = user's email address or blank
- Institute = user's institute abbreviation
- Product_ID = requested product types – each product type on a separate line
- Data_Release = data release; 0 indicates latest
- Start_Date = start date of requested data in yyyymmddhh
- End_Date = end date of requested data in yyyymmddhh
- Minimum_Latitude = minimum latitude of requested data
- Maximum_Latitude = maximum latitude of requested data
- Minimum_Longitude = minimum longitude of requested data
- Maximum_Longitude = maximum longitude of requested data
- 8_Day_Repeat = requested 8 day repeat cycle(s) -- each cycle on a separate line; -1 indicates none; 0 indicates all
- 8_Day_Tracks = requested 8 day repeat track(s) – each track on a separate line; -1 indicates none; 0 indicates all
- 91_Day_Repeat = requested 91 day repeat cycle(s) -- each cycle on a separate line; -1 indicates none; 0 indicates all
- 91_Day_Tracks = requested 91 day repeat track(s) – each track on a separate line; indicates none; 0 indicates all
- Output Directory = Output directory for data on remote host
- QL_flag = flag indicating quick look request or not; 0 = no, 1 = yes; only meaningful for special requests

Things to note:

- There should be no missing keywords and the only keyword that may have a blank value is for the email address.
- If no 91 data are available, then 91_Day_Repeat and 91_Day_Tracks should both be -1.
- Is the institute MSCF? If so, data will not be transferred to the output directory; it will be deposited in /SCF/dist/MSCF and you will have to move the files manually.
- The release: the latest releases of data are in /SCF/product_sets/current and previous releases are in /SCF/product_sets/previous. There are subdirectories under each of these directories indicating laser and reference orbit (i.e. L1_1102). The data request software loops over these subdirectories to fulfill the request. If the release is 0 then the software only loops over subdirectories under /SCF/product_sets/current. If the release is not 0, then subdirectories under /SCF/product_sets/previous are also checked.
- Pay attention to the latitudes, longitudes, start and end times, cycles, and tracks. If the request appears to be large (i.e. request is for the entire world, or for all cycles, all tracks) verify with the requestor that this is indeed what he/she wants. Then bearing in mind the disk space allotment in /SCF/tmp, either submit the entire request or break it up into separate requests by cutting, pasting, and modifying the email into separate emails.

When you are ready to submit the request, just forward the email to:

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

IMPORTANT: Forward the mail as text only. Forwarding the mail as html and text will result in duplicate entries and crash the request.

It is a good idea to monitor the request. For subscriptions, the only thing that happens is that the MYSQL database tables are updated with the subscription information. Verify that the following tables have been updated:

- SUBSCRIPTION_USER
- REQUEST_PRODUCT_SEG
- REQUEST_TRACKS
- REQUEST_CYCLES

For special requests, the data are processed in a temporary directory under /SCF/tmp. To find the directory:

- Type “cd /SCF/tmp”
- Type “ll –tr” to list files in reverse time order (newest files at the bottom of the list)
- Cd to the last “dir_” directory listed using the abbreviation for the site that the request was going to (i.e. dir_ALT_24722)

Check the files in the temporary directory to make sure that product files are being produced. Typing “ll –tr” will show the size of the latest file being produced. Upon doing this several times, if the file size is changing then data are still being processed and nothing is wrong; the process is just slow. If the request is large, creating products and all ancillary files may take a while. If little data are available at the mSCF, then the request will be finished very quickly, since the bulk of the data will need to be requested from the I-SIPS. Check the /SCF/dist/*rscf* (i.e.

/SCF/dist/ALT) directory to see if a PDR and XFR have been created. A log file with the user's name should be present listing the files created and/or requested from the I-SIPS.

NOTE: if the rSCF ID is MSCF, then the data will not be transferred out of /SCF/dist/MSCF. This is because MSCF stands for the main SCF itself which is not a remote site. Therefore, if the data need to go to a /SCF/product_sets subdirectory they must be moved manually. For any other rSCF ID, data will be transferred automatically to the remote site and moved to the specified subdirectory.

When the request has been fulfilled with mSCF data you should get an email like the example below:

```
/SCF/bin/ops/data_email.txt
Special request r0222 has been fulfilled with data from mSCF
```

If you don't get an email, if the request does not seem to be working, or if you get an error email, troubleshoot as per the "Troubleshooting Guide".

IMPORTANT NOTE: It is best to submit only one special request at a time. This is to prevent mmap errors from occurring (see section below, "Mmap Errors").

Special Addendum for Release 24 (GSAS V5.0)

The product record lengths have changed for GLA05-15 in release 24. This means that products of release 24 and below cannot be processed together. To that end:

- Release 24 and greater products exist in the /SCF/product_sets/current directories
- Release 23 and lesser products exist in the /SCF/product_sets/previous directories

- <u>To fulfill special requests specifying release >= 24 or release=0 (latest release):</u>
  - Forward email to `scf@icesat0.gsfc.nasa.gov`

- <u>To fulfill special requests specifying release <= 23:</u>
  - The shared library path needs to be changed and the Fortran routines need to be recompiled under V4.3 in the /SCF/bin/working directory.
  - To do this In the /SCF/bin/working directory:
    - In scf_environ.ksh change export SHLIB_PATH=/SCF/lib/V5.0 to export SHLIB_PATH=/SCF/lib/V4.3
    - In Makefile change SRCVERS=200510.0 to SRCVERS=200505.2 and LIBVERS=V5.0 to LIBVERS=V4.3
    - Type "make clean" then once it's done type "make"
  - Forward email to `scftest2@icesat0.gsfc.nasa.gov`
  - Once the request has been fulfilled, change the libraries back in /SCF/bin/working:
    - In scf_environ.ksh change export SHLIB_PATH=/SCF/lib/V4.3 back to export SHLIB_PATH=/SCF/lib/V5.0
    - In Makefile change SRCVERS=200505.2 to SRCVERS=200510.0 and LIBVERS=V4.3 to LIBVERS=V5.0
    - Type "make clean" then once it's done type "make"

SECTION 9.2: SUBMITTING QA UPDATES

Users can submit a product QA update request using the "Product QA" button on the GLAS SCF website. An email is sent to scf@icesat0.gsfc.nasa.gov where the software populates the QA_PRODUCT_UPDATE table in the ops database with the request parameters. An email is then sent to members of the CCB and to scf_manager@icesat0.gsfc.nasa.gov. The CCB must approve or deny the request before further action is taken.

Below is an example of the QA update request email:

```
/SCF/tmp/dir_15553/email_message2
Product QA update needs CCB approval to be submitted to I-SIPS

QA update request ID: 13

Product=gla08
Release=18.0
Date1=2003/02/29
Date2=2003/03/06
User=Kris
Site=MSCF
QA_update=Failed
Description=Saturated waveforms
```

Below is a breakdown of the request email components:

- QA update request ID = request ID number from database
- Product = product to be updated
- Release = data release
- Date1 = start date of data to be updated in yyyy/mm/dd
- Date2 = end date of data to be updated in yyyy/mm/dd
- User = user's name
- Site = user's institute abbreviation
- QL_update = Passed, Failed, or Inferred Passed
- Description = reason for QA update

When the CCB has made its decision about the request, you can update the database and submit the update to the I-SIPS by running the following code:

run_submit_qa.ksh <request_id> <ccb_accept>

where ccb_accept = y (yes) or n (no)

If you forget the syntax, just type run_submit_qa.ksh and it will show you the arguments needed. Once done, you will see a message like:

```
Changed ccb_accept value of U to Y for QA request ID = 13

Wrote /SCF/dist/ISIPS/MSCF.13.QAUF
```

If the CCB did not accept the request you will only see the top line.

It is a good idea to now verify that the QA_PRODUCT_UPDATE table has been updated.

If the CCB accepted the request, then the QAUF has to be transferred to the I-SIPS. The QAUF is like a POR in that it contains parameters that the I-SIPS ingests. However, once submitted nothing comes back to the mSCF. It is transferred with an XFR file, like the PDR.

Cd to the /SCF/dist/ISIPS directory to see the QAUF and XFR. To transfer the QAUF to the I-SIPS, type:

```
/SCF/src/perl/ops/run_scripts_push_isips.ksh
```

Check that the XFR is gone. You will have to remove the QAUF itself since there is no cleanup code for this.

SECTION 9.3: MODIFYING AND CANCELLING SUBSCRIPTIONS

Users need to notify mSCF personnel when they need a subscription modified, inactivated, or cancelled since these changes are done in the MYSQL database.

Inactivating Subscriptions:

If a user only wants to temporarily stop a subscription, it may be inactivated in the SUBSCRIPTION_USER database table, by changing the “status” to “N” (not active). To reactivate the “status” can be changed back to “A” (active) at any time. The process_subs.tcl script checks this status before fulfilling subscriptions.

Cancelling Subscriptions:

If a user definitely knows that he wants to cancel a subscription and never reactivate it, you may delete the subscription entirely from the database. While inactivating the subscription will effectively cancel it, deleting a subscription avoids later confusion and keeps the database clean. Deleting anything from the ops database must be done with extreme care.

To delete a subscription from the database, first be crystal clear on the subscription number that should be deleted. The user must tell you the subscription number that he wants cancelled. Check in the USER table that the subscription number matches the user that is requesting the cancellation and any criteria that the user has for it. If it does not, clarify the number again with the user.

Once the subscription number has been verified, just filter the data by “requestId” to match the subscription number and delete those rows from the following tables:

- SUBSCRIPTION_USER
- REQUEST_PRODUCT_SEG
- REQUEST_TRACKS
- REQUEST_CYCLES

Modifying Subscriptions:

Sometimes for small changes, it is easier for you to modify a subscription in the database than it is for the user to resubmit it. Parameters that can be changed easily by you in the database include:

- geographic range
- output directory
- start/end times
- product/segment
- cycles in same reference orbit

Note that if the geographic range is changed then the previously selected tracks may not match anymore, unless all tracks were requested.

If the user wants to modify reference orbit or tracks he must submit a new subscription for the change since the software calculates the tracks available in the reference orbit.

To modify an existing subscription, the user must tell you the subscription number that he wants modified and the exact changes that he wants to make. Check in the USER table that the subscription number matches the user that is requesting the modification and any criteria that the user has for it. If it does not, clarify the number again with the user.

Once the subscription number has been verified, filter the data by “requestId” to match the subscription number and edit the changing parameters in the appropriate tables:

- SUBSCRIPTION_USER table
    - output_dir
    - minLat
    - maxLat
    - minLon
    - maxLon
    - startDate
    - endDate
    - startJ2000 – use the Tools page of the SCF website to convert modified startDate to startJ2000
    - endJ2000 – use the Tools page of the SCF website to convert modified endDate to endJ2000

- REQUEST_PRODUCT_SEG table
    - product
    - segment

- REQUEST_CYCLES
    - begin_cycle
    - end_cycle

SECTION 10: MONITORING DATA AND REQUESTS

Since the data processing and request fulfillment is automated, the operator should not have to interact with it unless a problem arises. However, the operator should monitor the data received from the I-SIPS daily to ensure that it is complete. The operator should also periodically monitor requests to ensure that they are being fulfilled.

Monitoring Data received from the I-SIPS

The SCF expects to receive 14 revs of data from the I-SIPS daily when running a new laser period. During reprocessing of old laser periods, however, much more data may come at once. Also, several laser periods may be run at the same time.

The names of the files received along with subscription and energy analysis status information are recorded in the CREATION database table. The data processing code is in three parts: storing data, fulfilling subscriptions, and creating energy analysis plots for the SCF website. It is important to periodically check the /SCF/product_sets/tmp directory to make sure that data are moving to the storage directory. The script, **/SCF/bin/ops/test_check_subs_ea.ksh** should be run for a particular laser period to check the subscription and energy analysis status.

To run: test_check_subs_ea.ksh L3A <laser period>
i.e. test_check_subs_ea.ksh L3A

<u>Example output</u>:
laser=L3A
Subscription Status
D, 166
W, 150
Y, 2498

Energy Analysis Status
D, 1760
Y, 1054
errorCode=0

<u>Understanding the output</u>:

The number of files is given for each processing status indicator:
Y means “yes, completed”
N means “no, not completed”
NF means “no, not completed, forced to run a partial product set”
P means “processing”
D means “don’t need to do” (for example, no subscriptions are done on GLA03-4 and energy analysis is only run on GLA01)
W means “waiting, until get a full product set of data”
B means “bad, not processed” (for example, GLA04 files have no locations and are therefore not processed until BNA04 files are copied from BNL02 files. Except for these files, “bad” indicates a corrupt file that needs to be checked into.)

At the end of processing, all statuses should be “Y” or “D”.

Data are processed in 14 rev product sets for efficiency. However, at the beginning and end of laser periods or repeat cycles, the product set is not complete, so a “partial” product set of data must be run. **/SCF/bin/ops/test_check_pending_subs.ksh** and **test_check_pending_ea.ksh** can be run to change the status of any outstanding jobs so that they are set to run again.

Resetting Subscription Runs

Situation:
- Sometimes the subscription process may have a problem or get hung. If the process has run for a few hours and nothing seems to be happening, kill all subscription processes and then run the script **/SCF/bin/ops/test_check_pending_subs.ksh** to reset the file statuses in the MySQL database.
- Sometimes data have not come for a while and files with a “W” status have not had subscriptions run on them because they are a part of an incomplete 14 rev product set. You can force the files to be run by running **/SCF/bin/ops/test_check_pending_subs.ksh** to reset the file statuses in the MySQL database.

To run: test_check_pending_subs.ksh <reset option> <laser period>
i.e. test_check_pending_subs.ksh 7 L3A

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

Understanding the reset options:
- Up to 4 subscription processes may run at once, hence P1, P2, P3, P4. Either all processing statuses may be reset or any individual one.
- Files with status “N” will be run only if all files in the 14 rev product set are available. The status must be “NF” to force a run on a partial product set.

What it does: Resets any “P”, “B”, or “W” jobs to “N” or “NF” (depending on option used) in the MySQL database so these files can be run through the subscription program again.

Once the jobs are reset the subscription program comes in automatically via cron every minute. Use **test_check_subs_ea.ksh** to check that the files are being processed.

Resetting Energy Analysis Runs

Problem: Sometimes the energy analysis process may have a problem or get hung. If the process has run for a few hours and nothing seems to be happening, kill all energy analysis processes and then run the script **/SCF/bin/ops/test_check_pending_ea.ksh** to reset the file statuses in the MySQL database.

To run: test_check_pending_ea.ksh

What it does: Resets any “P” jobs to “N” in the MySQL database so these files can be run through the energy analysis program again.

Once the jobs are reset the energy analysis program comes in automatically via cron every minute. Use **test_check_subs_ea.ksh** to check that the files are being processed.

Monitoring Requests from the rSCF's

When a request email is received, it must be forwarded as text only to scf@icesat0.gsfc.nasa.gov. An email from populate_db.tcl will be sent when the request has been received and populated in the database. Special request information is stored in the following database tables: SPECIAL_REQUEST_USER or SUBSCRIPTION_USER, REQUEST_PRODUCT_SEG, REQUEST_TRACKS, and REQUEST_CYCLES. Files created to fulfill the request are stored in the DISTRIBUTION_FILES table. Another email is sent when the special request has been fulfilled with data from the mSCF specifying whether the request was fulfilled with data or not. If it hasn't been fulfilled, check over the request for an obvious reason and notify the requestor if necessary.

SECTION 11: AUTOMATED PROCESSING

To make the mSCF as automated as possible, data processing and system monitoring scripts are either run at set time intervals via cron jobs or triggered by the receipt of an email to a specific account.

SECTION 11.1: EMAIL-TRIGGERED PROCESSES

The following scripts are triggered by an email to a specific account:

Account	Process Triggered	Scripts Run
scf	/SCF/bin/ops/readMail.ksh	parse_mail.tcl, populate_db.tcl, data_select.tcl
scftest2	/SCF/bin/working/readMail.ksh	parse_mail.tcl, populate_db.tcl, data_select.tcl
scfstat	/SCF/bin/ops/readStatMail.ksh	parse_stat_mail.tcl
scfistat	/SCF/bin/ops/readIStatMail.ksh	parse_istat_mail.tcl

Below are the general functions of the triggered processes:

readMail.ksh	- Receives special request: updates database tables and fulfills request
readStatMail.ksh	- Receives I-SIPS Oracle report and sends error email if mSCF does not have all files listed in the report
readIStatMail.ksh	- Receives instrument commands and target of opportunity information: updates database tables with information

For more detailed descriptions of the above scripts refer to the *SCF Data Request Software Detailed Design Document.*

SECTION 11.2: CRON PROCESSES

The following is the current crontab under the scf account:

```
* * * * * /SCF/src/perl/ops/run_scripts_pull_isips.ksh
* * * * * /SCF/src/perl/ops/run_scripts_pull_browse.ksh
* * * * * /SCF/src/perl/ops/run_scripts_alt.ksh
* * * * * /SCF/src/perl/ops/run_scripts_browse.ksh
* * * * * /SCF/src/perl/ops/run_scripts_buf.ksh
* * * * * /SCF/src/perl/ops/run_scripts_lidar.ksh
* * * * * /SCF/src/perl/ops/run_scripts_mit.ksh
* * * * * /SCF/src/perl/ops/run_scripts_nsidc.ksh
* * * * * /SCF/src/perl/ops/run_scripts_lrscf.ksh
* * * * * /SCF/src/perl/ops/run_scripts_osu.ksh
* * * * * /SCF/src/perl/ops/run_scripts_simpl.ksh
* * * * * /SCF/src/perl/ops/run_scripts_ucsd.ksh
* * * * * /SCF/src/perl/ops/run_scripts_utcsr.ksh
* * * * * /SCF/src/perl/ops/run_scripts_wff.ksh
* * * * * /SCF/src/perl/ops/run_scripts_scfweb.ksh
* * * * * /SCF/src/perl/ops/run_scripts_mscf.ksh
* * * * * /SCF/src/perl/ops/test_scripts_acc.ksh
* * * * * /SCF/src/perl/ops/test_scripts_special.ksh
* * * * * /SCF/src/perl/ops/run_cleanup.ksh >> /SCF/tmp/cleanup.txt 2>&1
* * * * * /SCF/bin/ops/run_process_data.ksh >>
/SCF/tmp/cron_proc_data_errors.txt 2>&1
* * * * * /SCF/bin/ops/run_process_subs.ksh >>
/SCF/tmp/cron_proc_subs_errors.txt 2>&1
* * * * * /SCF/bin/ops/run_process_ea.ksh >> /SCF/tmp/cron_proc_ea_errors.txt
2>&1
* * * * * /SCF/bin/ops/run_acc_process_data.ksh >>
/SCF/tmp/cron_acc_proc_data_errors.txt 2>&1
00 * * * * /SCF/bin/ops/run_check_dist.ksh >> /SCF/tmp/cron_dist_errors.txt
2>&1
00,30 * * * * /SCF/bin/ops/run_check_ps.ksh >> /SCF/tmp/cron_ps_errors.txt
2>&1
00,30 * * * * /SCF/bin/ops/run_check_ps_rscf.ksh >>
/SCF/tmp/cron_psr_errors.txt 2>&1
00,30 * * * * /SCF/bin/ops/run_check_ps_linux_rscf.ksh >>
/SCF/tmp/cron_psr_errors.txt 2>&1
00 02,19 * * * /SCF/bin/ops/run_daily_cleanup.ksh >>
/SCF/tmp/cron_daycl_errors.txt 2>&1
00 03 * * * /SCF/bin/clean_scf_tmp.ksh >> /SCF/tmp/cron_cleantmp_errors.txt
2>&1
#00 16 * * * /SCF/bin/ops/run_stat_report.ksh >>
/SCF/tmp/cron_statrep_errors.txt 2>&1
30 01,06,12,18 * * * /SCF/bin/ops/run_prod_trends.ksh >>
/SCF/tmp/cron_prod_trends_errors.txt 2>&1
00 02 * * * /SCF/src/web/ops/run_subscription_display.ksh >>
/SCF/tmp/cron_sub_dis_errors.txt 2>&1
00 05 * * * /SCF/bin/ops/run_too.ksh >> /SCF/tmp/cron_too.txt 2>&1
00 10,18 * * * /SCF/src/web/ops/run_istats_display.ksh >>
/SCF/tmp/cron_istats_dis_errors.txt 2>&1
```

The first five columns dictate the time that the process will be run at:

minute The minute of the hour, 0-59
hour The hour of the day, 0-23
monthday The day of the month, 1-31
month The month of the year, 1-12
weekday The day of the week, 0-6, 0=Sunday

An asterisk (*) indicates “all” for that field. (i.e. * * * * * means once per minute of every hour of every day of every month). The number sign (#) indicates that the line is commented out and therefore not activated.

Below are the general functions of the automated processes:

/SCF/src/perl/ops/run_scripts_pull_isips.ksh – Pulls data from the I-SIPS to the mSCF
/SCF/src/perl/ops/run_scripts_pull_browse.ksh – Pulls data from the I-SIPS to the mSCF
/SCF/src/perl/ops/run_scripts_[rSCF].ksh – Pushes data to the rSCF
/SCF/src/perl/ops/run_scripts_mscf.ksh – Moves data from ingest cache on the mSCF to the appropriate subdirectory where they are staged for processing or distribution
/SCF/src/perl/ops/test_scripts_acc.ksh – Moves acctest data from ingest/ACC directory on the mSCF to the acctest directory where they are staged for processing
/SCF/src/perl/ops/test_scripts_special.ksh – Moves special ops data from ingest/ACC directory on the mSCF to the acctest directory where they are staged for processing
/SCF/src/perl/ops/run_cleanup.ksh – Cleans up transferred data if successful POR has been received
/SCF/bin/ops/run_process_data.ksh – Processes and stores new I-SIPS data
/SCF/bin/ops/run_process_subs.ksh – Fulfills subscriptions with new I-SIPS data
/SCF/bin/ops/run_process_ea.ksh – Creates energy analysis plots with new I-SIPS data
/SCF/bin/ops/run_acc_process_data.ksh – Processes and stores new acctest data
/SCF/bin/ops/run_check_dist.ksh – Checks if files in certain directories are over x hours old
/SCF/bin/ops/run_check_ps.ksh – Checks if number of processes exceeds a threshold and if certain directories are almost out of disk space
/SCF/bin/ops/run_check_ps_rscf.ksh – Checks if number of processes exceeds a threshold and if certain directories are almost out of disk space on rSCF’s
/SCF/bin/ops/run_check_ps_linux_rscf.ksh – Checks if number of processes exceeds a threshold and if certain directories are almost out of disk space on Linux rSCF’s
/SCF/bin/ops/run_daily_cleanup.ksh – Determines if certain files may be deleted
/SCF/bin/clean_scf_tmp.ksh – Removes old directories in /SCF/tmp.
/SCF/bin/ops/run_stat_report.ksh – Creates and emails status report (commented out except during laser-on periods)
/SCF/src/web/ops/run_subscription_display.ksh – Posts subscription data from database onto SCF website
/SCF/bin/ops/run_too.ksh – Compares TOOs with GLA06 data
/SCF/src/web/ops/run_istats_display.ksh - Posts instrument and target of opportunity data from database onto SCF website

For more information on editing the crontab file and on the data transfer scripts refer to the *SCF Interface Software Operator's Guide for mSCF.* For more detailed descriptions of the data

processing and monitoring scripts refer to the *SCF Data Request Software Detailed Design Document.*

SECTION 11.3: PROCESS MESSAGES

There are several scripts that run via cron jobs to monitor the system. For time to time, these scripts will send messages to mSCF personnel if the situation arises. These scripts include:

check_dist.tcl:

Example of email message:

```
/SCF/bin/ops/check_dist.txt
Current date and time = 2003-05-15 09:00:01

The following files are more than 4 hours old:

/SCF/dist/ISIPS/MSCF.r0222.POR 2003-05-14 16:07:37
```

This message indicates that files having been sitting around for longer than the allotted amount of time. To change the threshold number of hours or directories searched, edit run_check_dist.ksh. To determine what to do depends on which files are listed:

- POR's in /SCF/dist/ISIPS: the request to I-SIPS has not be fulfilled in x hours. Inform the I-SIPS operators by sending an email to ops@isipspr1.nascom.nasa.gov or by submitting a Mantis problem report. If you know that the request was large or that the I-SIPS is busy or down you may want to wait a little longer before contacting them.

- Files in /SCF/ingest/ISIPS: Either a lot of data is being downloaded or something happened. Check the PDR to see how many files are being downloaded. Typing "ll –tr" will show the size of the latest file being produced. Upon doing this several times, if the file size is changing then data are still being downloaded and nothing is wrong; the process is just slow. If nothing is changing then the downloading may have stopped. You should have gotten an email indicating a problem. If not, check the cron log (see "Troubleshooting Guide"). If there was an error downloading the files send an unsuccessful PAN to the I-SIPS (see the "Sending a PAN to the I-SIPS" section). If there was an error moving the files to their subdirectory, touch the XFR to try the move again.

- Files in /SCF/dist/*rSCF* or in /SCF/ingest on a remote site: Something probably happened to the data transfer to the remote site. Check the cron log (see "Troubleshooting Guide") to see. If this is the case, then all the data are probably still in the /SCF/dist/*rSCF* directory on icesat0. Just touch the XFR for each old PDR in the directory (be aware that newer PDR's may be in the directory as well, but do not create XFR's for these since they may already be in the process of transferring).

- Files in /SCF/product_sets/tmp: If the PDR's and XFR's are still in this directory is usually means that an updated rev file needs to be obtained from the I-SIPS. Refer to the "Rev File" section. If some XFR's are gone and process_data.tcl is running, it could just mean that a lot of data need to be processed and it is taking a while.

- Files in /SCF/product_sets/GLA04: These are non-LPA GLA04 files (0002.DAT-0006.DAT). They may be deleted.

check_ps.tcl:

Example of email message:

```
/SCF/bin/ops/check_ps.txt
Current date and time = 2003-06-12 13:50:00

The number of scf processes is more than 200: 203
```

This message indicates that the number of processes is close to the system limit. To change the number of processes threshold edit run_check_ps.ksh. Get a systems administrator to help monitor the processes if you get this email. If the number of processes exceeds the system limit fork problems can occur (refer to the “Troubleshooting Guide” for steps to take if this happens).

Example of email message:

```
/SCF/bin/ops/check_ps.txt
Current date and time = 2003-05-13 22:40:00

Disk is 100% full in /SCF/ingest
```

This message indicates that available disk space is dangerously low in the listed directory. To change the threshold percent or directories searched, edit run_check_ps.ksh. Refer to the “Troubleshooting Guide” for steps to take when the disk space is full.

check_ps_rscf.tcl:

Example of email message:

```
/SCF/bin/ops/check_ps_rscf.txt
Current date and time = 2003-06-12 13:50:00

The number of scf processes is more than 200 at NSIDC: 203
```

This message indicates that the number of processes is close to the system limit at the remote site. To change the number of processes threshold edit run_check_ps_rscf.ksh file on that site. Get a systems administrator to help monitor the processes if you get this email. If the number of processes exceeds the system limit fork problems can occur (refer to the “Troubleshooting Guide” for steps to take if this happens).

Sometimes you will get an error message that looks like this:

```
/SCF/bin/ops/check_ps_rscf.txt
Current date and time = 2003-12-16 12:50:00

The number of scf processes is more than 200 at WFF:
ssh_exchange_identification:
```

This indicates that something has happened to the connection between the mSCF and the rSCF. If the number of processes is anything other than a number, it means that the automated ssh is not working. This happens often due to temporary network glitches. If you see error messages for an hour or more indicating that the connection to a specific site is down, try to ssh to the site manually. If you cannot connect, inform a systems administrator and email the

systems administrator at that site inquiring if their system is up. If you can ssh to the site, but need to type in the scf password for two successive logins, then inform a systems administrator because the automated ssh should not need the password entered.

Check_ps_rscf can send another type of email message:

```
/SCF/bin/ops/check_ps_rscf.txt
Current date and time = 2003-05-13 22:40:00

Disk is 100% full in at NSIDC: /SCF/product_sets/data
```

This message indicates that available disk space is dangerously low in the listed directory at the remote site. To change the threshold percent or directories searched, edit run_check_ps_rscf.ksh. When this happens a lock file is put in the site's distribution directory on the mSCF to indicate to the subscription code not to move data into there. The concern is that if a directory becomes full on a remote site, then the distribution disk may become full and inhibit any rSCF from receiving data. With this lock file in place, data is moved to /SCF/product_sets/tmp/dist for safekeeping. Email the users at the site to clean up their directories.

If process_subs.tcl could not move files into a distribution directory because a lock file was present, you should receive an error email with errorcode=50 at the bottom.

When the distribution directory is not longer full and the lock file is gone, the next time check_ps_rscf.tcl comes in it will remove the lock file. It will then check the file names in /SCF/product_sets/tmp/dist against the subscription numbers for the site in the database and move the correct files to the distribution directory. You should receive an email like this from mv_saved_files.tcl indicating that files are being moved:

```
/SCF/bin/ops/mv_saved_files.txt
STOP_DATA file removed for UCSD: Moving /SCF/product_sets/tmp/dist/*s0188* to
/SCF/dist/UCSD
STOP_DATA file removed for UCSD: Moving /SCF/product_sets/tmp/dist/*s0184* to
/SCF/dist/UCSD
STOP_DATA file removed for UCSD: Moving /SCF/product_sets/tmp/dist/*s0189* to
/SCF/dist/UCSD
STOP_DATA file removed for UCSD: Moving /SCF/product_sets/tmp/dist/*s0250* to
/SCF/dist/UCSD
STOP_DATA file removed for UCSD: Moving /SCF/product_sets/tmp/dist/*s0191* to
/SCF/dist/UCSD
```

Parse_stat_mail.tcl:

This code compares an email from the I-SIPS Oracle database listing the files processed that day against the files in the SCF data directories. If a file was not found at the SCF then an email is sent.

Example of email message:

```
/SCF/tmp/dir_21104/stat_mail.txt

The following files completed processing at the I-SIPS but are not at the
mSCF:
```

```
GLA04_012_2103_002_0097_0_01_0002
GLA04_012_2103_002_0097_0_01_0003
```

This message indicates that the files listed were processed by the I-SIPS but were not found in either the ingest or data directories on the mSCF. Take these steps to find the files:

- Check the following directories again to see if the files were transferred between the time the mail was written (usually 6 am) and now:
  - /SCF/ingest/ISIPS (the ingest directory)
  - /SCF/product_sets/tmp (the directory where files are stored until they are processed)
  - /SCF/product_sets/current/Lw_xxxx where w is the laser number and xxxx is the reference orbit (the directory where files are stored after they are processed)
- If they are not in the above directories, grep on the cron_ISIPS log files in the /SCF/tmp directory to see if the files were transferred from the I-SIPS.
- If there is no record in the log files that the files were transferred, notify the I-SIPS personnel at ops@isipspr1.nascom.nasa.gov or by submitting a Mantis problem report saying that the mSCF never received the files.

Parse_istat_mail.tcl:

This code updates the instrument database tables displayed on the SCF website. The two tables are RTSCM_UPDATE (Real-time Saved command) and TOO_UPDATE (Target of Opportunity). The code automatically updates the tables when an email is received by the scfistat account on icesat0.

When the RTSCM_UPDATE table is updated successfully, you will receive the following email:

Subject: Mail from parse_istat_mail.tcl - RTSCM update

```
Date: Fri, 6 Jan 2006 08:00:01 -0500 (EST)
      >From: isf@osb1.wff.nasa.gov
Message-Id: <200601061300.k06D01WJ003995@glasist2.gsfc.nasa.gov>
To: scfistat@icesat0.gsfc.nasa.gov
Subject: /glasist/isf/outgoing/scfcsm_log

  2005/041-14:21:18.00 CMD ICE GGPCBIAS1
  2005/041-14:21:20.00 CMD ICE GGCDBIAS1
  2005/041-14:21:22.00 CMD ICE CBMCNTSET with COUNT 19
  2005/041-14:21:24.00 CMD ICE CBMEXE with INDEX 120
```

Since the subject of this message indicates an “update”, it is OK to just ignore and delete it from your mailbox. If the subject is “Error from parse_istat_mail.tcl” then you may have to look into why the database update failed.

When the TOO_UPDATE table is updated successfully, you will receive the following email:

Subject: Mail from parse_istat_mail.tcl - TOO update

```
Begin forwarded message:
```

```
> From: ICESat FDS <icefds@navstar.colorado.edu>
> Date: November 17, 2005 6:41:34 PM EST
> To: moc.ice_science@lasp.colorado.edu
> Subject: TOO Update for Load Period 2005/323-12:00:00 to
> 2005/324-12:00:00
> Reply-To: ICESat FDS <ice.fds@lasp.colorado.edu>
>
>
> PATH - ACCEPTED
> ;North_Rim
> 15517
> 2005/323-15:04
> 36.79000/247.98218
> 35.79000/247.80782
> Glas bias commands added: YES
```

Since the subject of this message indicates an “update”, it is OK to just ignore and delete it from your mailbox.

Sometimes you may get an email indicating that there was no update. That may be OK. If the format is not like that of the TOO (PATH - …) then it was another type of email that we don’t populate our database with. For example, this email can just be ignored and deleted:

Subject: Mail from parse_istat_mail.tcl - No update

```
Date: Fri, 6 Jan 2006 01:09:50 -0500 (EST)
      >From: Oracle DHAPS Server <dhaps@osb1.wff.nasa.gov>
Message-Id: <200601060609.k0669o7S013608@glasist2.gsfc.nasa.gov>
To: scfistat@icesat0.gsfc.nasa.gov
Subject: maneuver.txt

Maneuver    0 on 021219000000.0000, Loc= A, dV=    0.000000 m/s, yaw=    0.00
deg, da=       0.0000 m, di=     0.000000 deg
Maneuver    1 on 030211033511.6443, Loc= D, dV=    0.045818 m/s, yaw=    0.00
deg, da=      84.4599 m, di=     0.000000 deg
Maneuver    2 on 030219121144.7947, Loc= A, dV=    0.052690 m/s, yaw=    0.00
deg, da=      97.1258 m, di=
```

Sometimes you may get an email indicating an error, but it still may be OK to just ignore and delete the message if the format of the message doesn’t match the TOO pattern, The message below is an example of this:

Subject: Error from parse_istat_mail.tcl

```
Target_Status.txt File for FDS Quaternion Software
=====================================================

Initiated by: iceacs 20051122113000 20051123123000 251 MISSION
Date at start of run:
Mon Nov 21 17:10:25 GMT 2005

PROCESSED TARGETS OF OPPORTUNITY:
PATH 15560 3.68651 114.08548 4.68651 113.94864 1
```

```
PATH 15560 -75.49078 261.50274 -76.49361 260.19131 1
PATH 15561 19.93667 261.00316 18.93667 260.85796 1
PATH 15568 8.78150 280.09507 9.78150 279.95677 1


OMITTED TARGETS OF OPPORTUNITY:
NO OMITTED TARGETS OF OPPORTUNITY
```

Therefore, only if the email subject says “Error” or “No update” and the message has the format of the TOO, do you need to investigate why the TOO_UPDATE table was not populated with the information in the email. To test the email again, simply forward it to

scfistat@icesat0.gsfc.nasa.gov.

Daily_cleanup.tcl:

Example of email message:

```
/SCF/bin/working/daily_cleanup_error.txt
Files have been moved to /SCF/product_sets/deleted_files from
/SCF/product_sets/current/L2_2103
```

This message indicates that files were moved from /SCF/product_sets/current/RELEASE_12 to /SCF/product_sets/deleted_files because they were determined to be ready for removal. Files can be removed if they meet one of two criteria:

- The use_flag in the INSTRUMENT_UPDATES table indicates “N” for the time of the file
- A newer version of the file exists

Files are moved to /SCF/product_sets/deleted_files rather than being deleted to ensure that data files are not mistakenly removed. Note that UR and PS files are moved along with GL files. Before removing the files from this directory, spot check a few of the files first:

- Search for the next version of the file in the directory it came from
- If a newer version exists, remove the files in /SCF/product_sets/deleted_files
- If no newer version exists, check the use_flag in the INSTRUMENT_UPDATES table for the time of the file
- If the use_flag indicates “N”, remove the files in /SCF/product_sets/deleted_files
- If the use_flag indicates “Y”, investigate why the files were moved

SECTION 12: TROUBLESHOOTING GUIDE

SECTION 12.1: DATA TRANSFER PROBLEMS

If an error occurs during a data transfer, the PAN will indicate a problem, but an email will be sent explaining the problem in further detail. The cron.log.### file may also show a problem. Use the error description in the email to troubleshoot the problem. Once the problem has been solved, all the files must be put into the output cache again so that they will be transferred when the cron job invokes. Since the scripts reference the complete PDR, it is important that all of the files listed in the PDR are available to be transferred again, even if some of them were already transferred before the transfer was interrupted. In most cases, the XFR file may have been removed, since this is the first thing that most of the scripts do. The scripts read the PDR or PAN only if the XFR is also present, and therefore remove the XFR first so that another cron job does not try to access the same PDR or PAN at the same time. Therefore, if a transfer has to be run again, if the XFR is missing, it needs to be recreated. This can be accomplished simply by using the "touch" command, since the XFR only needs to exist, not necessarily contain anything. The XFR file name is simply the PDR or PAN file name with the extension ".XFR" added at the end. For example: "touch filename.PDR.XFR".

Below are common error messages generated by the Perl scripts and a general outline of troubleshooting techniques if an error occurs. Refer to the *SCF Interface Software Detailed Design Document* for detailed descriptions of the PDR, XFR, and PAN files.

Error Type	**Check that:**
Error creating file	- Disk space is available * - Permissions allow write access **
File does not exist	- Full file name is correct - Permissions allow read access **
Error opening file	- Permissions allow read access ** - Full file name is correct
Cannot create directory	- Disk space is available * - Permissions allow write access **
Number of files has not been read from PDR	- PDR follows correct format
Number of files does not match number in PDR	- PDR follows correct format - Disk space is available * - All files listed in PDR exist in the directory - File transfer was complete
PDR is for a subscription and number of files is 0	- PDR follows correct format - PDR is really for a subscription
Checksum for file does not match checksum in PDR	- PDR follows correct format - File checksum is accurate - File transfer was complete
PDR verification did not pass all tests	- PDR follows correct format - All files listed in PDR exist in the directory - File checksums are accurate - File transfer was complete
PAN file does not indicate successful transfer	- Look for accompanying email for error details
Expected data can not be found	- Cron job ran

*** If disk space is not available:** Have system administrator create more disk space on system
**** If file has restrictive permissions:** Have system administrator change permissions on file

If you have received an email indicating an error during a file transfer, follow these guidelines to troubleshoot the problem:

- The cron job on the mSCF pulls data from the I-SIPS and pushes data to the rSCF's. Therefore if a file transfer fails, you have to check the cron log file on the mSCF.
- The cron job on the rSCF only moves data from the ingest directory to a subdirectory. If this fails, check the cron log on that rSCF.
- "Error during scp" messages may have indicated that the network was down, but transfer may have worked once the network was up again.
- The /SCF/product_sets/subdirectory that a file should go into is based on one of two things:
  - If the file was part of a product set fulfilling a request to an rSCF, the subdirectory was selected by the user in the data request GUI and can be seen in the SPECIAL_REQUEST_USER database table.
  - If the file was from the I-SIPS, the subdirectory is determined by the script /SCF/src/perl/version/get_subdir.pm based on data type and file type.

Example: If you receive an email indicating an error with a file transfer from the I-SIPS to the mSCF:

- On icesat0, check /SCF/ingest/ISIPS to see if the file and PDR are there
- If there are no files in /SCF/ingest/ISIPS, they may have already been moved to /SCF/product_sets/tmp or /SCF/product_sets/current.
- Look at the PDR file if present
  - Check for correct file types and data types
  - Check for correct file names and checksums
- On icesat0, check the /SCF/tmp/cron.log.### file for information on whether the files were transferred or not. You may need to do a grep on the date or a "ll –tr cron*" to determine which cron log covers the date and time in question.
- Email the I-SIPS operators at ops@isipspr1.nascom.nasa.gov or submit a Mantis problem report specifying to touch XFR again so the PDR and files will be transferred again.

Example: If you receive an email indicating an error with a file transfer from the mSCF to the rSCF:

- On icesat0, check /SCF/dist/rSCF to see if the file is there
- Check /SCF/ingest/rSCF to see if a PAN is there for the transfer
- Log onto the rSCF and check /SCF/ingest to see if the file is there
- If there are no files in /SCF/ingest, they may have already been moved to a subdirectory under /SCF/product_sets.
- Look at the PDR file if present
  - Check for correct file types and data types
  - Check for correct file names and checksums
- On icesat0, check the /SCF/tmp/cron.log.### file for information on whether the files were transferred or not. You may need to do a grep on the date or a "ll –tr cron*" to determine which cron log covers the date and time in question. For a description of which scripts write to which cron log files refer to the *SCF Interface Software Operator's Guide for mSCF.*
- If the PDR in /SCF/dist/rSCF is present, touch the XFR again so PDR and files will be transferred again.

SECTION 12.2: DISK FULL PROBLEMS

How you will know:
- You may receive an email indicating an error
- Data will not be moving

What to do:
These sorts of errors can be quite messy to deal with since they usually result in partially transferred and/or processed data. Therefore it is not always sufficient to just restart the transfer or processing, but to determine at what point the stoppage took place and work from there.

Disk is full in /SCF/tmp:
This usually means that too many temporary directories have accumulated, especially if processing had stopped so that they contain large product files. It can also occur if a data request is too large to be fulfilled in one request.

When this happens:
- Run /SCF/bin/clean_scf_tmp.ksh either as user "scf" or "root". This will remove "dir_" directories under /SCF/tmp over one week old. If this doesn't help, just remove all the directories by typing "rm –r –f dir_*".
- If this happens due to a large request, after removing the temporary directory, submit the request again, however break the email into separate requests either by time or geographic region or by product.

Disk is full in /SCF/ingest:
This means that the SCF has downloaded so much data from the I-SIPS that the ingest cache was overwhelmed. This means that any transfers from the I-SIPS will have stopped mid-stream and the PDR's will probably not be verified since at least one file in each transfer will be incomplete.

When this happens:
- Stop the cron job pulling data from the I-SIPS:
    - To edit the cron file: *crontab –e* (under scf account)
    - Comment out command running /SCF/src/perl/ops/run_scripts_pull_isips.ksh (It is the first line) by putting a # at beginning of the line.
- Contact Suneel and tell him that the SCF is temporarily not accepting data.
- Note that data will not be able to be transferred until the /SCF/ingest directory has some space. This is because the incoming PAN's go into this directory.
- You may have to move some data to another directory for temporary storage to clear some space. GLA07 files are good because they are so large.
- Touch XFR's for PDR's in /SCF/ingest/ISIPS so any PDR's that were transferred before or after the stoppage can be processed. If PDR is not complete or any file checksum is incorrect, you will receive an email.
- Establish which PDR's are incomplete (they are the ones left in the directory when all the data have been processed). Removed the files associated with them because you need to receive them again. Either send Suneel a list of PDR's that you need again or create unsuccessful PAN's for these PDR's (see the "Sending a PAN to the I-SIPS" section) so the files will be transferred again. You can also just run the script /SCF/bin/ops/test_touch_isips_pdr.ksh. You may need to first modify the ISIPS_DIR and

INGEST_DIR variables in the script depending on which ingest directory has unfinished PDR's. This script will list the PDR's in the ingest directory and touch XFR files for them in the outgoing directory on the I-SIPS host.

- When the /SCF/ingest/ISIPS directory has cleared out, resume the cron job pulling data from the I-SIPS.
- Contact Suneel and tell him that the SCF is accepting data again.
- Periodically monitor the disk space and directory to ensure that data are moving.

Disk is full in /SCF/dist:
This either means that /SCF/dist/acctest is full or that the SCF is processing and distributing data more quickly than it is being transferred to the rSCF's from the distribution caches. This means that any subscriptions running will have stopped at the point that the data are moved to the distribution cache (which, of course, is at the end). Also, any transfers from the I-SIPS that have files go directory to a distribution cache (i.e. GLA03-4, ANC files) will have stopped mid-stream and the PDR's will probably not be verified since at least one file in each transfer will be incomplete.

When this happens:

- Stop the cron job running subscriptions:
    - To edit the cron file: *crontab –e* (under scf account)
    - Comment out command running /SCF/bin/ops/run_process_subs.ksh by putting a # at beginning of line
- Check that /SCF/dist/acctest directory. If it has a lot of data, especially old releases, ask David Hancock and John DiMarzio which data may be removed.
- Touch XFR's for PDR's in each /SCF/dist/*rSCF* directory that has outstanding PDR's. Don't do too many at once or else you may run into secure shell entropy errors (refer to the "Fork/Entropy Problems" section). Periodically monitor the disk space and directories to ensure that data are moving. Continue to touch PDR's until all data are distributed.
- When the /SCF/dist directories have cleared out resume the cron job running subscriptions.
- You may need to reprocess any FN files that bombed due to the lack of space. Refer to the "When Subscription Data are Not Processed" section on how to do this.

Disk is full in /SCF/product_sets/current/L#:
This means that the SCF has processed so much data that the storage directory was overwhelmed. This means that any data processing will have stopped at the point that the data are moved to the /SCF/product_sets/current directory.

When this happens:

- Stop the cron job running process_data:
    - To edit the cron file: *crontab –e* (under scf account)
    - Comment out command running /SCF/bin/ops/run_process_data.ksh by putting a # at beginning of line
- Since data is not transferred out of /SCF/product_sets/current, one of the following needs to happen to create more disk space:
    - A systems administrator needs to add more disk space to /SCF/product_sets
    - Data has to be backed-up and removed from /SCF/product_sets. Get approval before doing this.
- When disk space is restored resume the cron job processing data.
- Periodically monitor the disk space and directory to ensure that data are moving.

Disk is full in /SCF/product_sets/tmp:
This means that SCF data processing in the /SCF/product_sets/tmp directory could not keep up with the rate of data download from the I-SIPS. Any data movement from the /SCF/ingest/ISIPS directory to the /SCF/product_sets/tmp directory will have stopped mid-stream.

When this happens:

- Stop the cron job moving data from /SCF/ingest/ISIPS to /SCF/product_sets/tmp:
    - To edit the cron file: *crontab –e* (under scf account)
    - Comment out command running /SCF/src/perl/ops/run_scripts_mscf.ksh by putting a # at beginning of line
- You may have to move some data to another directory for temporary storage to clear some space. GLA07 files are good because they are so large.
- Let the data processing software continue so hopefully it will continue to move data to storage.
- When you have adequate space, bring back the files you moved out. Touch XFR's for PDR's so those files are processed.
- You may need to reprocess any FN files that bombed due to the lack of space. Refer to the "When Subscription Data are Not Processed" section on how to do this.
- When all the FN files are processed, resume the cron job moving data from /SCF/ingest/ISIPS to /SCF/product_sets/tmp.
- Check for .tmp directories in /SCF/product_sets/tmp. If any exist, then "rm –r –f .tmp*" to remove them. If you don't then you may error out when trying to move the data. Touch the XFR's for PDR's in /SCF/ingest/ISIPS. If a PDR is not complete or any file checksum is incorrect, you will receive an email.
- If an error occurs, it may be because some of the files in the PDR are in /SCF/ingest/ISIPS while others are in /SCF/product_sets/tmp. Move these files back to /SCF/ingest/ISIPS and retouch the XFR.
- Periodically monitor the disk space and directory to ensure that data are moving.

SECTION 12.3: FORK/ENTROPY/PRNG PROBLEMS

How you will know:
- Data will not be moving

What to do:
The first thing to do if data does not appear to be moving is to check the cron logs in /SCF/tmp. Grep on “fork”, “entropy”, or “PRNG” or simply display the latest cron_ISIPS.log.### file and scan for errors. These sorts of errors can be quite messy to deal with since they usually result in partially transferred and/or processed data. Therefore it is not always sufficient to just restart the transfer or processing, but to determine at what point the stoppage took place and work from there.

Fork:
“Fork” errors indicate that there were too many processes running at the same time. All scripts whether they were transferring or processing data are temporarily halted, but resume once the number of processes sinks back below the threshold. This means that in many cases PDR’s are not verified since the transfer of one file may have been affected.

When this happens:
- Let the load die down until few processes are running.

Entropy:
“Entropy” errors indicate that there were too many secure shell processes running at the same time. All scripts transferring data from the I-SIPS or to the rSCF’s are halted until the entropy daemon is restarted.

When this happens:
- Get a systems administrator to restart the entropy daemon.
- Touch XFR’s for PDR’s in each /SCF/dist/*rSCF* directory that has outstanding PDR’s. Don’t do too many at once or else you may get the entropy error again. Periodically monitor the disk space and directories to ensure that data are moving. Continue to touch PDR’s until all data are distributed.
- If transfers from the I-SIPS were halted, determine which PDR’s in /SCF/ingest/ISIPS are incomplete and create unsuccessful PAN’s for these PDR’s (see the “Sending a PAN to the I-SIPS” section) so files will be transferred again.
- If for some reason they can’t do this, you should at least process the files you have. To do this you will need to determine for each PDR the files that exist and edit the PDR’s accordingly. Touch the XFR’s so the files will be further processed.

PRNG:
“PRNG is not seeded” errors indicate that there were too many secure shell processes running at the same time. All scripts transferring data from the I-SIPS or to the rSCF’s are temporarily halted until the number of processes dies down. This means that after a while data may continue to be transferred, but some PDR’s may not be verified because intermittant files may have had errors during their transfers.

When this happens:
- Touch XFR’s for PDR’s in each /SCF/dist/*rSCF* directory that has outstanding PDR’s. Don’t do too many at once or else you may get the PRNG error again. Periodically monitor the

disk space and directories to ensure that data are moving. Continue to touch PDR's until all data are distributed.

- If transfers from the I-SIPS were halted, determine which PDR's in /SCF/ingest/ISIPS are incomplete and create unsuccessful PAN's for these PDR's (see the "Sending a PAN to the I-SIPS" section) so files will be transferred again.
- If for some reason they can't do this, you should at least process the files you have. To do this you will need to determine for each PDR the files that exist and edit the PDR's accordingly. Touch the XFR's so the files will be further processed.

SECTION 12.4: MMAP ERRORS

How you will know:

- You may receive an email indicating an error
- Data processing will have stopped

The error message will look like this:

```
Error in running mkbinrev
/usr/lib/dld.sl: Call to mmap() failed - BSS /SCF/lib/V4.3/libprod.sl
/usr/lib/dld.sl: Not enough space
```

What to do:

This error occurs when several requests and/or subscriptions are running at the same time, access the libraries simultaneously, and conflict. Libraries are used when the UR, PS, BN, and GR files are created (in the above case, the process making the BN files died). To prevent getting this error, it is best to never have more than two subscriptions and/or special requests running at the same time. Since subscriptions are processed automatically and unwittingly, it is prudent to only submit one special request at a time.

- If this error occurs on a special request, simply resubmit it by resending the email to the scf account.

- If this error occurs on a subscription, restage the data to be processed again. See the section below, “When Subscription Data are not Processed” for details on how to do this.

SECTION 12.5: WHEN A SPECIAL REQUEST IS NOT FULFILLED

How you will know:

- You may receive an email indicating an error
- Someone from a rSCF may notify you

Synopsis of special requests:

When someone at an rSCF submits a special request through their data request GUI, an email is sent to the mSCF where it is parsed into request parameters. Product sets are produced based upon those request parameters and put into that rSCF's distribution cache along with a PDR for file transfer.

What to do:

- If you know the path of the tmp directory, cd to there or do the following to find the directory:
  - Type "cd /SCF/tmp"
  - Type "ll –tr" to list files in reverse time order (newest files at the bottom of the list)
  - Cd to the last "dir_" directory listed using the abbreviation for the site that the request was going to (i.e. dir_ALT_24722)

- The following are expected files in this directory: A short summary for each which help you determine which ones to look at to troubleshoot the problem:
  - ctrl_pdr.dat: input file to create the delivery PDR
  - errors.txt: errors from parse_mail.tcl; just lists email parameters if no errors
  - errors_out.txt: errors from populate_db.tcl; ignore "can't find package tbcload" error and "errorInfo-can't read "conn": no such variable" error
  - errors_sr.txt: errors from data_select.tcl
  - input_files.txt: input files to data_select
  - kris_Sep12_15:39:41_information.errors: errors file sent to rSCF
  - mail_parameters.txt: email parameters
  - new_tmp_dir.txt: name of renamed tmp directory
  - orgTrack.txt: input tracks to data_select
  - out.txt: output from parse_mail.tcl
  - output.txt: output from populate_db.tcl
  - output_sr.txt: output from data_select.tcl
  - passID.txt: list of processed pass ID's
  - passid_time.bin: intermediate file for data_select
  - perl_output.txt: output from create_pdr.pl
  - requestIdFile.txt: request ID

Start with viewing errors_sr.txt as most of the errors occur here. Output_sr.txt is also helpful for seeing exactly what happened.

- Notice if any output files were created (REQ, GLA).
  - If an REQ file is 600 bytes, there is no data in it, only headers.
  - Check sizes of GLA files – if they are increasing over multiple "ll" listings then the processing is not finished, just slow.
  - If there are no output files in the tmp directory, check the distribution cache – they may have already been moved there, ready for transfer.

- If the processing appears to have stopped somewhere between REQ file creation and product creation, you may need to rerun data_select, parse_req, or prod_create from the command line in the tmp directory to see the errors.

When you find the problem, depending on where the processing failed, you may be able to create the files or you may need the remote site to submit the request again (or you can do it for them). Any files created need to be placed in the remote sites distribution cache along with a PDR and an XFR. To create these manually, refer to the *SCF Interface Software Operator's Guide for mSCF.*

SECTION 12.6: WHEN DATA PROCESSING STOPS

How you will know:

- You may receive an email indicating an error

Synopsis of subscription data processing:

When product sets are received from the I-SIPS, they are moved to /SCF/product_sets/tmp. They are then moved to /SCF/product_sets/tmp/distID where they are renamed with a PID extension and the associated data management system files are created. When finished they are moved to /SCF/product_sets/current. In the CREATION database table, the columns run_subs and run_ea are set to "N".

What to do if receive email message:

```
/SCF/tmp/process_data_errors.txt

Wed Jun 11 17:05:00 EDT 2003 Invoking process_data.tcl
file=GLA02_011_1102_009_0101_0_01_0001.DAT
check_pid.tcl: files in FN 456.51507 not all in same PID
errorInfo=
errorCode=11
err_file=/SCF/tmp/process_data_errors.txt
```

This means that the FN file in /SCF/product_sets/tmp lists files in different 14 rev product sets. Copy the FN file to one or more extra files making sure to use unique names (i.e. FN456.51507-02.txt, FN456.51507-03.txt) for each. Be sure to change the ID in each copied FN file to match the new file name. Cut and paste among the FN files so that files in the same product set are listed in each FN file.

What to do if a data processing stopped:

- Check /SCF/product_sets/tmp for the distID subdirectory: if it is there the processing stopped, if it is gone, then the processing finished.
- If the distID subdirectory is still there then rerun the subscription processing code:
  - Move GLA files from the subdirectory to /SCF/product_sets/tmp. Do "ls –l" to make sure that the files are not links. If they are remove them. The script save/rm_linked_files.ksh can help with this if there are many.
  - Type "rm –R *subdir*" to remove the subdirectory and its contents.
  - Check that all the files in the FN file are accounted for or create a new FN file if not available. The script /SCF/bin/ops/run_create_fn_files.ksh can help with this if there are many.
  - Touch the FN XFR. The script save/touch_fn_xfr.ksh can help with this if there are many.
  - Run test_process_data.ksh in the /SCF/bin/ops directory and watch for error output.

SECTION 12.7: WHEN SUBSCRIPTION PROCESSING STOPS

How you will know:

- You may receive an email indicating an error

Synopsis of subscription data processing:

The run_subs column in the CREATION database table is checked for files that are “N”.
The file parameters are compared with rSCF subscription parameters in the database. If they match, subsetted product sets are produced based upon those subscription parameters and put into that rSCF’s distribution directory along with a PDR for file transfer.

What to do if receive email message:

```
/SCF/tmp/process_subs_errors.txt
ERROR : -10001, 3, open_bin_head, Error Opening File for Input:
/SCF/product_sets/current/L3I/GRA04_029_2121_002_0155_0_01_0001.P1684
child process exited abnormally
    while executing
"exec $bin_path/data_select $tmp_dir/$ds_ctrl_file"
Error running run_ds_pc.tcl

sub_error.tcl - Subscriptions are being changed to 'B' for the following
files
file=GLA04_029_2121_002_0159_0_01_0001.P1684
file=GLA04_029_2121_002_0155_0_01_0001.P1684
file=GLA04_029_2121_002_0157_0_01_0001.P1684
file=GLA04_029_2121_002_0161_0_01_0001.P1684
file=GLA04_029_2121_002_0163_0_01_0001.P1684
file=GLA04_029_2121_002_0165_0_01_0001.P1684
file=GLA04_029_2121_002_0167_0_01_0001.P1684
errorCode=2
err_file=/SCF/tmp/process_subs_errors.txt
```

This means that since the BNL02 files were not available for the GLA04 files, they couldn’t have subscriptions filled against them. At the end of data processing, run the script test_check_pending_subs.ksh to set the files back to “N” so they can be rerun again, hopefully after the BNL02 files have arrived.

What to do if receive email message:

```
/SCF/tmp/process_subs_errors.txt
reqID=s0250
int_reqID=250
utmp_dir /SCF/tmp/dir_11179.1703.s0250
db_table=SUBSCRIPTION_TRACKS_2
sql_str=update CREATION set subs_run='Y'  where
ISIPS_file='GLA15_429_2121_002_0127_0_01_0001.P1682'
errorCode=50
err_file=/SCF/tmp/process_subs_errors.txt
```

ErrorCode=50 means that subscriptions were filled without error, but files were moved to /SCF/product_sets/tmp/dist because a remote site ran out of disk space.

What to do if a subscription processing stopped:

- A temporary directory is created in /SCF/tmp for each subscription (i.e. dir_3746.1703.s0220)
- Refer to the section “When a Special Request is not Fulfilled” for details on what the files mean.

What to do if a subscription has not been fulfilled:

- Check the MYSQL database to see what the subscription parameters are for the subscription ID (s#####).
- Check the MYSQL database to see if the files received match the subscription parameters.

SECTION 12.8: SENDING A PAN TO THE I-SIPS

When downloading data under normal conditions, the mSCF sends a PAN back to the I-SIPS indicating success or failure. Even if an error occurs, if an email was received than an unsuccessful PAN was probably sent to the I-SIPS. Only when unusual situations occur, such as disk space, fork, or entropy problems, does a PAN need to be manually made and sent to the I-SIPS. If any of these problems has created a situation where data need to be resent, then an unsuccessful PAN must be created so that the I-SIPS operators can recreate the XFR to allow the PDR to be retransferred. If data have been successfully received or you just don't need the data to be transferred again, then a successful PAN must be created so that the I-SIPS operators can cleanup their distribution cache.

In /SCF/ingest/test/PAN are two PAN templates: PDR543_89726.PAN is a successful PAN and PDR543_89729.PAN is a failed PAN. Copy the PAN you need to /SCF/dist/ISIPS and rename with the PDR you are creating the PAN for, then create the XFR. To transfer the PAN to the I-SIPS type:

```
/SCF/src/perl/ops/run_scripts_push_isips.ksh
```

Check that the XFR has been removed from the /SCF/dist/ISIPS directory. Once you are sure that the PAN has been transferred to the I-SIPS, delete the PAN.

APPENDIX A: A DAY IN THE LIFE CHEAT SHEET

Here's a list of things to do every day:

- Check and remove files from **/SCF/product_sets/GLA04**. These are granules 2-6 of GLA04 that we don't keep unless someone needs them.
- Check and remove files from **/SCF/product_sets/deleted_files**. These are files from when the laser is off or not producing data we want to keep.

When the SCF is receiving ops data from the I-SIPS:

- See the "SCF Programmer's Guide" section "Procedures for Before Data Reprocessing" for more information on preparing to receive data.

- Run the script**, /SCF/bin/ops/test_check_subs_ea.ksh <laser period>** to check the subscription and energy analysis status for a particular laser period.

- Periodically check the **/SCF/product_sets/tmp** directory to make sure that data are moving to the storage directory.

- When all the data have been received and processed, if any data have not been processed, run the scripts, **/SCF/bin/ops/test_check_pending_subs.ksh <reset option> <laser period>** and **test_check_pending_ea.ksh** to change the status of any outstanding jobs so that they are set to run again.

- See the "SCF Programmer's Guide" section "Procedures for After Data Reprocessing" for more information on what to do after data are received.

APPENDIX B: ABBREVIATIONS & ACRONYMS

GLAS	Geoscience Laser Altimeter System
GSFC	Goddard Space Flight Center
ICESat	Ice, Cloud, and land Elevation Satellite
ISF	Instrument Support Facility
I-SIPS	ICESat Science Investigator-led Processing System
LIDAR	LIght Detection And Ranging
MIT	Massachusetts Institute of Technology
mSCF	Main Science Computing Facility
NASA	National Aeronautics and Space Administration
OSU	Ohio State University
rSCF	Remote Science Computing Facility
UCSD	University of California at San Diego
UTCSR	University of Texas Center for Space Research
UW	University of Washington
WFF	Wallops Flight Facility