ICESat (GLAS) Science Processing Software Document Series

GSAS User's Guide
Version 5.0

Jeffrey Lee/Raytheon ITSS
Observational Science Branch
Laboratory for Hydrospheric Processes
NASA/GSFC Wallops Flight Facility
Wallops Island, Virginia 23337

August 2004

ICESat Contacts:

H. Jay Zwally, ICESat Project Scientist
NASA Goddard Space Flight Center
Greenbelt, Maryland 20771

Bob E. Schutz, GLAS Science Team Leader
University of Texas Center for Space Research
Austin, Texas 78759-5321

David W. Hancock III, Science Software Development Leader
NASA/GSFC Wallops Flight Facility
Wallops Island, Virginia 23337

Foreword

This document contains the GLAS Science Algorithm Software (GSAS) User's Guide. This document is developed under the structure of the NASA STD-2100-91, a NASA standard defining a four-volume set of documents to cover an entire soft-ware life cycle. Under this standard a section of any volume may, if necessary, be rolled out to its own separate document. This document is a roll- out of the user guide within the Product Specification Volume.

The GEOSCIENCE LASER ALTIMETER SYSTEM (GLAS) is a part of the EOS program. This laser altimetry mission will be carried on the spacecraft designated EOS ICESat (Ice, Cloud and Land Elevation Satellite). The GLAS laser is a frequency-doubled, cavity-pumped, solid state Nd:YAG laser.

This document was prepared by the Observational Science Branch at NASA GSFC/WFF, Wallops Island, VA, in support of B. E. Schutz, GLAS Science Team Leader for the GLAS Investigation. This work was performed under the direction of David W. Hancock, III, who may be contacted at (757) 824-1238, David.W.Hancock@nasa.gov (e-mail), or (757) 824-1036 (FAX).

This document was created through the efforts of the GLAS Science Software Development Team. Current team members include:

Raytheon/Kristine Barbieri

Raytheon/Suneel Bhardwaj

Raytheon/Lisa Brittingham

Raytheon/John Dimarzio

972/David W. Hancock, III

Raytheon/Peggy Jester

Raytheon/Jeffrey Lee

Raytheon/Dennis Lockwood

Raytheon/Steve McLaughlin

SSAI/Steve Palm

Raytheon/Carol Purdy

Raytheon/Lee Anne Roberts

Raytheon/Jack Saba

Table of Contents

List of Figures

List of Tables

Section 1
Introduction

1.1 Identification of Document

This document is identified as the GLAS Science Algorithm Software (GSAS) User's Guide. The unique document identification number within the GLAS Ground Data System numbering scheme is TBD. Successive editions of this document will be uniquely identified by the cover and page date marks.

1.2 Scope of Document

The GLAS I-SIPS Data Processing System, show in Figure 1-1, provides data processing and mission support for the Geoscience Laser Altimeter System (GLAS). I-SIPS is composed of two major software components - the GLAS Science Algorithm Software (GSAS) and the Scheduling and Data Management System (SDMS). GSAS processes raw satellite data and creates EOS Level 1A/B and 2 data products. SDMS provides for scheduling of processing and the ingest, staging, archiving and cataloging of associated data files. This document is the User's Guide.

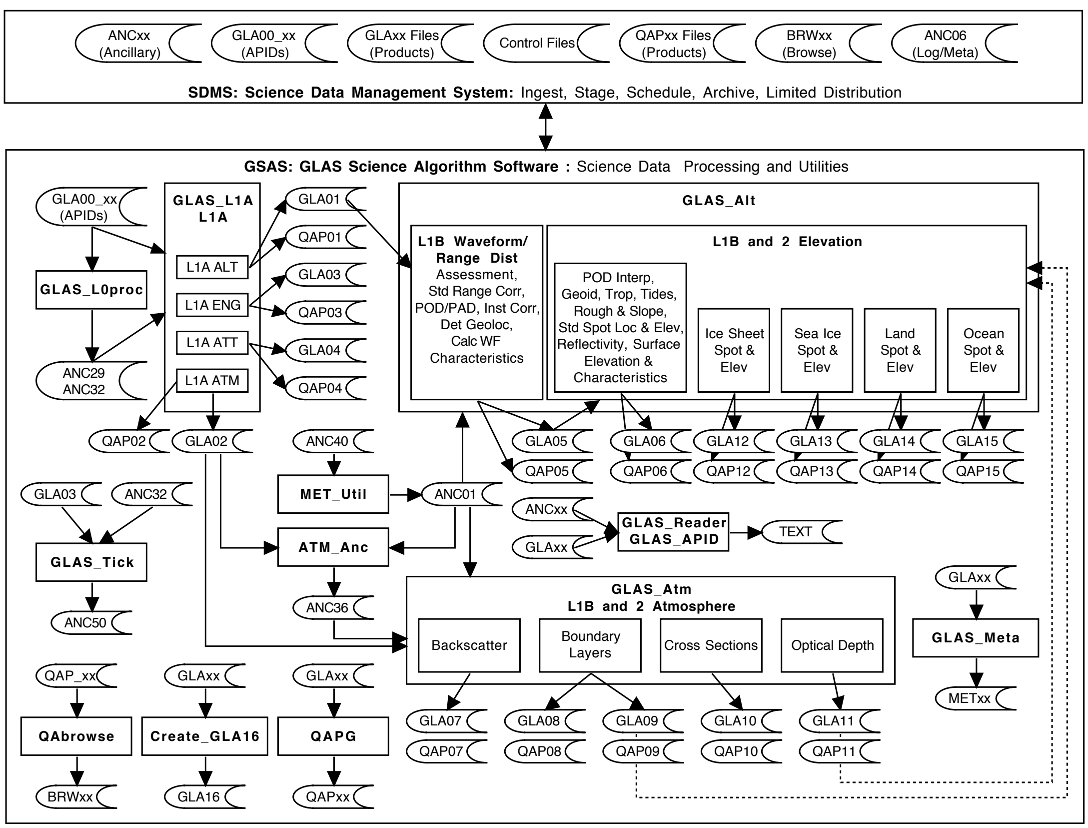

Figure 1-1 I-SIPS Software Top-Level Decomposition

1.3 Purpose and Objectives of Document

The purpose of this document is to provide the GSAS end users with instructions explaining how to operate the software effectively.

1.4 Document Organization

This document's outline is assembled in a form similar to those presented in the NASA Software Engineering Program [Information Document 2.3a].

1.5 Document Change History

Table 1-1 Document Change History

Document Name: GLAS Science Algorithm Software Users' Guide		
Version Number	**Date**	**Nature of Change**
Version 0	August 1999	Original Version.
Version 1	November 2000	Revised for V1 software.
Version 2	November 2001	Revised for V2 software.
Version 2.2	July 2002	Revised for V2.1 and 2.2 software.
Version 4.0	October 2002	Revised for V3.0 software.
Version 5.0	August 2004	Revised for V4.0 software.

Section 2

Related Documentation

2.1 Parent Documents

Parent documents are those external, higher-level documents that contribute information to the scope and content of this document. The following GLAS documents are parent to this document.

a) *GLAS Science Software Management Plan* (GLAS SSMP), August 1998, NASA/GSFC Wallops Flight Facility, NASA/TM-1999-208641/Version 3/Volume 1.

The GLAS SSMP is the top-level Volume 1 (Management Plan Volume) document of the four volumes of NASA software engineering documentation [Applicable Reference 2.2c]. It dictates the creation and maintenance of the Product Specification Volume (Volume 2). This document is a roll out of the Product Specification Volume.

2.2 Applicable Documents

a) *NASA Software Documentation Standard Software Engineering Program*, NASA-STD-2100-91, July 29, 1991, NASA.

b) *GLAS Science Algorithm Software (GSAS) Detailed Design Document* (GSAS DD), Version 4.0, August 2004, NASA/GSFC Wallops Flight Facility.

c) *GLAS Science Algorithm Software (GSAS) Version Description Document* (GSAS VDD), Version 4.0, August 2004, NASA/GSFC Wallops Flight Facility.

d) *GLAS ISIPS Operational Procedures Manual*, TBD.

2.3 Information Documents

a) *GLAS Standard Data Products Specification - Level 1*, Version 7.0, August 2004, NASA/GSFC Wallops Flight Facility.

b) *GLAS Standard Data Products Specification - Level 2*, Version 7.0, August 2004, NASA/GSFC Wallops Flight Facility.

c) *GLAS Science Data Management Plan* (GLAS SDMP), NASA/TM-1999-208641/Ver.4/Vol.2, July 1999, NASA/GSFC Wallops Flight Facility.

Section 3
Overview

3.1 Purpose

GSAS generates the GLAS Standard Data Products and associated metadata describing the products and their quality. The software uses GLAS telemetry and ancillary data to produce the products using algorithms defined by the GLAS Instrument and Science Teams.

GSAS is delivered as a set of libraries and PGEs (Product Generation Executables). A PGE is an executable program which performs a specific function. The 'core' PGEs perform specific portions of the GLAS data processing and generate deliverable GLAS Data Products (Products). The core PGEs are accompanied by a set of utility PGEs which perform such functions as creating ancillary data files, performing quality assurance and generating browse products.

Throughout this document, files are referenced as one of two types: GLA or ANC. GLA files are fixed-length, integer-binary format product files containing Level 0-2 GLAS science data. Level 3 science data is provided with the HDF-EOS formatted GLA16. GLA files are both input and output to GSAS. ANC files are multi-format ancillary files which are required for processing. A list of the GLA and ANC files is supplied in Appendix A. These files are detailed in the GLAS Data Management Plan and GLAS Standard Data Product Specifications Documents.

3.2 Environment

GSAS software is developed for and delivered on the UNIX platform. This document assumes that the reader is familiar with UNIX operating system conventions. The software is currently developed, tested and supported on HP/UX 11.0, Fortran 2.5.10 and RSI IDL 6.0.

3.3 Functions

The functionality of the PGEs is grouped into four major categories: L1A processing, Altimetry processing, Atmosphere processing, and utility functions.

L1A processing uses Level-0 data and L1A science algorithms to create Level-1A products (GLA01-04).

Altimetry processing uses L1A data and waveform/elevation science algorithms to create Level-1B waveform (GLA05) and Level-1B/2 elevation (GLA06, GLA11-15) products.

Atmosphere processing uses L1A data and atmosphere science algorithms to create Level 1-B and 2 (GLA07-11) atmosphere products.

The utilities perform functions which are ancillary to the main data processing process. This includes such tasks as data reformatting, ancillary file creation, browse product creation, and data validation.

In order to perform these functions, PGEs are designed to reflect a natural division in GLAS data processing. Table 3-1 lists the name, category, and function of each of these executables.

Table 3-1 GSAS PGEs

PGE	Category	Functionality
GLAS_L0proc	Utility	Creates an ancillary file from Level-0 APIDs which contains timing and data alignment information.
GLAS_L1A	L1A	Reads the GLAS_L0proc ancillary file and associated APID files to create Level-1A GLAS data.
GLAS_Alt	Altimetry	Reads altimetry-related Level-1A data files to create Level-1B and 2 GLAS altimetry files.
GLAS_Atm	Atmosphere	Reads atmosphere-related Level-1A data files to create Level1b and 2 GLAS atmosphere files.
GLAS_Reader	Utility	**Non-Production**. Reads specified GSAS ancillary and/or product files and creates a text-based, human-readable representation of the data.
GLAS_Meta	Utility	Reads specified product files and creates EOS-compliant metadata inventory files.
GLAS_GPS	Utility	**Non-Production**. Uses the externally-supplied RINEX software to process GPS data. (this utility was developed before UTexas assumed the GPS processing role.)
GLAS_APID	Utility	**Non-Production**. Reads APID files and creates column-based text output suitable for importing into spreadsheets.
GLAS_Tick	Utility	Reads GLA03 and ANC32 files to create ANC50 files containing engineering statistics and GPS update event data
atm_anc	Utility	Reads atmosphere Level-1 products and creates atmosphere calibration files.
met_util	Utility	Reads and subsets standard Meteorological files, creating GSAS ancillary MET files.
reforbit_util	Utility	Reads Reference Orbit files to create ascending equatorial crossings.
createGran_util	Utility	Reads Predicted Orbit files, creating all ascending equatorial crossings, and +/- 50 degree latitude crossings.
create_regionsMASK_util	Utility	
QAbrowse	IDL Utility	Creates graphical browse products from product QA files.

Table 3-1 GSAS PGEs (Continued)

PGE	Category	Functionality
qapg	Utility	Creates comparative QAP files for GLA product files.
NOSE_util	Utility	Creates a output file containing NOSE off-nadir track and segment coordinates for ESDIS and track and segment numbers for GLAS product headers and metadata.
prod_util	Utility	**Non-Production.** A collection of utilities which provide such functions as writing sample product files for testing purposes (product_test), subsetting GLA00 APIDs (stripper), providing ESDIS-deliverable sample product readers (gsas_prod_readers), and stripping extraneous information from spacecraft-test 1984 APID data (strip1984).

The GSAS design allows a significant amount of commonality among the executables. This is beneficial from both a software maintenance and ease-of-use point of view. Common functionalities include:

- Use of a control file for dynamic processing information.
- Ability to perform partial processing.
- Ability to perform selective processing based on input and output time specification.
- Use of change-controlled ancillary files which contain changeable Science Team-supplied parameters.
- Input data time synchronization
- Creation of metadata which contains a full processing history.
- Use of a standardized error/message reporting facility with user-defined options available.

3.4 Time Specification

All GSAS times are specified in J2000 UTC, with an epoch of January 1, 2002 12:00:00. In some cases, GPS times are used internally, but are converted to J2000 UTC before presentation to the user in order to preserve consistency. All user-interaction with GSAS requires time specifications in J2000 UTC.

3.5 Restrictions and Limitations

GSAS has the following limitations:

- GSAS is supported on HP/UX 11.0 with HP Fortran 2.5.10 and RSI IDL 6.0.

Refer to the appropriate version of the GSAS Version Description document for additional information.

Section 4

Installation and Initialization

This section will detail installation instructions for GSAS. As GSAS will be delivered with HP/UX 11.0 binaries, recompilation is not required, but recommended. Additionally, instructions are provided in case the user wishes to recompile using debug or optimization options. Recompilation is not supported on different architectures since GSAS has only been tested under HP/UX 11.0/F90 2.5.10.

Note: Throughout this document, the text refers to directory names preceded by a "$" and typed in all capital letters. This means that the directory path is local to the specific installation and not hard-coded within the GSAS. (This document uses standard UNIX 'sh' variable conventions to indicate such.) Other pathnames are specified relative to the local directories and only the relative relationship is important.

4.1 Unpack the Tarfile

Create a production directory for the GLAS_Exec program. This directory is designated $GLAS_HOME.

```
mkdir $GLAS_HOME
```

Copy the delivered tar file into GLAS_HOME and untar it. We will refer the tar file as 'gsas.tar', but the actual delivery file may be named differently. Designate $DIST_DIR as the source of the gsas.tar tarfile.

```
cd $GLAS_HOME
cp $DIST_DIR/gsas.tar .
tar xf gsas.tar
```

The untar process will create the directory structure described in Table 4-1.

Table 4-1 GSAS Directory Structure

Directory	Description
$GLAS_HOME/bin	GSAS executables. Executables should always be linked from this directory.
$GLAS_HOME/lib	GSAS shared libraries. (The user should always set the SHLIB_PATH environmental variable to this directory or link the libraries from this directory into the working directory.)
$GLAS_HOME/cc_util	Contains makefile utilities.
$GLAS_HOME/idl	Contains IDL code.
$GLAS_HOME/docs	Contains PDF versions of released documents.
$GLAS_HOME/data	Contains sample control and ancillary files.

Table 4-1 GSAS Directory Structure

Directory	Description
$GLAS_HOME/src	Contains Fortran90 anc C source code.
$GLAS_HOME/test	Contains test data and scripts to verify proper operation of the software.

4.2 Compilation

Change to the production directory and perform the following makes:

```
cd $GLAS_HOME
make clean
make runtime
make install
```

After this process is complete, all object files, libraries, and executables will have been re-created and installed in the appropriate directories. This is the recommended mode of compilation. It is the method used to create the distribution binaries and for final integration and acceptance testing.

4.2.1 Debugging

The debug option signals that the compiler should produce code which performs bounds checking, floating point exception checking, runtime error trapping and allows for use of a symbolic debugger.

If debug checking is desired, the software must be recompiled with the "debug" flag.

```
cd $GLAS_HOME
make clean
make debug
make install
```

After this process is complete, all object files, libraries, and executables will have been re-created with the debug option and installed in the appropriate directories.

4.2.2 Optimization

The optimization option signals that the compiler should produce optimized code. These optimizations include such things as full level-3 optimization, automatic parallelization, array reordering, and subroutine inlining. Note that compiling with optimization takes a significant amount of time.Given that optimization is a "risky" process, refer to the appropriate version of the GSAS Version Description Document for additional information.

If optimization is desired, the software must be recompiled with the "fast" flag.

```
cd $GLAS_HOME
make clean
make fast
make install
```

After this process is complete, all object files, libraries, and executables will have been re-created, optimized with the "fast" option and installed in the appropriate directories.

4.3 Verification

GSAS includes a suite of data and scripts to verify proper operation of the software. This suite is a subset of the GSAS Acceptance test. The test runs each of the GSAS PGEs with approximately 600 seconds of test data. The test should take less than 20 minutes to complete.

Assuming the software has been successfully compiled and installed, verify its operation as follows:

```
cd $GLAS_HOME/test
./verify.sh
```

The following shows a successful verification test. Contact the GSAS development team if your results differ

```
GSAS Verification Test

Cleaning up...

Testing GLAS_L0proc...

  Passed.

Testing GLAS_L1A...

  Passed.

Testing reforbit_util...

  Passed.

CreateGran_util...

  Passed

Testing met_util...

  Passed.

Testing atm_anc...

  Passed.

Testing GLAS_Atm...

  Passed.

Testing GLAS_Alt (Waveforms)...

  Passed.

Testing GLAS_Alt (Elevation)...
```

Passed.

After successful completion, the newly created data may be removed by running the following command:

```
cd $GLAS_HOME/test
./cleanup.sh
```

Section 5

Common Functionality

GSAS code was designed to maximize software reuse capabilities. As such, it provides for a great deal of common functionality among its PGEs. GSAS uses standardized control files, constants files, and error/status reporting facilities. This makes GSAS easier to use since, for example, a control file for one PGE is basically the same style as a control file for other PGEs.

5.1 Control Files

Control files are the interface between GSAS and the user (or controlling process). Control files provide dynamic control information to PGEs. Static information is provided by ancillary constants files (ANC07). A single control file can provide instructions for more than one PGE. Section breaks are used to delimit those instructions of significance to the specific PGE.

5.1.1 Use

The name of the control file is passed as a command-line argument during each invocation of the PGE. Most PGEs will terminate with a fatal error if the command-line argument is missing, the specified file does not exist, or the file is unreadable. The exceptions to this rule are GLAS_Reader and GLAS_APID, which provide a rudimentary user-interface when invoked without a control file name.

5.1.2 Format

GSAS control files are designed to be part of a larger control file used by one or more PGE. As such, it specifies sections that identify the PGE that will perform the task requiring the inputs contained in the section. Each section is bounded by an "=" sign in column 1, followed by the PGE name that requires the control inputs. Exact section names will be shown in the PGE-specific control file section of this document.

All GSAS control files are created in standard GSAS “keyword=value” format. This format is text-based and consists of a line containing a keyword/value pair delimited by an equal sign (=). The ordering of the keywords is not relevant but should follow a convention for consistency. Multiple instances of certain keywords are allowed. The keyword is not case sensitive. Spaces are allowed, but not required. Comment lines must be prepended by a “#” character. The keyword and value are limited to 255 characters each.

5.1.3 Content

PGE sections within a control file contain both common and process-specific information. The process-specific portions of control files will be provided within the documentation for each specific executable. This section will document common elements of the PGE control files.

Within a control file section, some information is required, other is optional. Required single-instance keywords include:

Table 5-1 Required Single-Instance Keywords

Keyword	Value
TEMPLATE_NAME=	Name of the control file template.
EXEC_KEY=	Unique (per day) execution key
DATE_GENERATED=	Date the control file was generated.
OPERATOR=	Operator who generated the control file.
PGE_VERSION=	External Version number of the target PGE.

Optional multiple-instance keywords include:

Table 5-2 Optional Multiple-Instance Keywords

Keyword	Value
PASSID=	Pass-related information
INPUT_FILE=	Input filename and associated information
OUTPUT_FILE=	Output filename and associated information
WRITE_CONST=	Signals that the specified constants should be written to ANC06. (described in Section 5.3.3.)

5.1.4 PASSID Specification

A PASSID section is required in the control file when creating GLA products. There should be one instance of the following keyword/values for all tracks which fall within the minimum/maximum time of the data being processed. This information is required for GLAS_L1A, GLAS_Alt, and GLAS_Atm. This information is NOT required for GLAS_L0proc or other utilities.

```
PASSID=revolution_num<sp>passid<sp>start_time<sp>stop_time<
sp>equator_crossing_lon<sp>nose_path_number.
```

Descriptions of the PASSID elements are provided in Table 5-3.

Table 5-3 PASSID Control Line Elements

Element	Description
revolution_num	integer, containing the auto-incrementing rev number.
passid	11-byte character, further described below.
start_time	double-precision float, containing J2000 UTC time in seconds.
stop_time	double-precision float, containing J2000 UTC time in seconds.

Table 5-3 PASSID Control Line Elements (Continued)

Element	Description
equator_crossing_lon	float, containing the equator crossing longitude.
nose_path_number	integer, containing the NOSE path number.

The eleven-byte passid field will be treated as follows: prkkccctttt. Descriptions of each element are provided in Table 5-4.

Table 5-4 passid Field Description

Field	Description
p	repeat ground track phase (integer, length=1)
r	reference orbit number (integer, length=1)
kk	instance (integer, length=2)
ccc	cycle (integer, length=3)
tttt	track (integer, length=4)

Example (with nonsense values):

```
#
#---------Pass ID Information
#
PASSID=107 11010010001 85300.0 85350.0 35.0 37
PASSID=108 11010010001 85351.0 85400.0 45.0 38
PASSID=109 11010010002 85401.0 85500.0 55.0 39
PASSID=110 12010010003 85501.0 85600.0 65.0 40
PASSID=111 12010010004 85601.0 85700.0 75.0 41
PASSID=112 22010010005 85701.0 85800.0 85.0 42
PASSID=113 22010020006 85851.0 85900.0 95.0 43
PASSID=114 22020020007 85901.0 86000.0 105.0 44
PASSID=115 22020020007 86001.0 86500.0 115.0 45
```

5.1.5 Input/Output File Specification

Input and Output files are required to be designated using the GSAS-standard naming convention defined in Appendix A. The type of each file specified is determined by parsing specific components of the filename which are required by all of the naming methods defined in the specification. These common components of all filenames are:

```
HHHxx_mmm...ffff.eee
```

(where: HHH is the type identification, xx is the type id number, mmm is the release number, ff is the file sub-type, and eee is the file extension.)

GSAS software uses the type identification, the type id number and the file sub-type to determine what type of file is specified in the control file. The filetype-parsing routines are not case-sensitive when determining the type of file specified. However, the filenames are case-sensitive during file opening and creation.

All files are required to be delimited by start and stop times. These times are floating point values specified on the control line as J2000 time in seconds. On both input and output, records are skipped until the time in the current record is greater than or equal to the specified start-time and less than or equal to the specified stop-time. Static ancillary files are required to have start-times and stop-times present for consistency, but these are currently ignored.

The minimum required formats for input and output file specifications are:

```
INPUT_FILE=file_name<sp>start_time<sp>stop_time
OUTPUT_FILE=file_name<sp>start_time<sp>stop_time
```

Additionally, GLA product file entries should contain segment and version information. This information is specified in the format:

```
INPUT_FILE=file_name<sp>start_time<sp>stop_time<sp>gran_rel
_num<sp>gran_ver_num<sp>gran_segment
```

```
OUTPUT_FILE=file_name<sp>start_time<sp>stop_time<sp>gran_re
l_num<sp>gran_ver_num<sp>gran_segment
```

Segment and version information fields are described in Table 5-5.

Table 5-5 File Segment and Version Fields

Field	Description
gran_rel_num	granule release number (CCB controlled, mmm in filenaming convention.) Character max length of 20.
gran_ver_num	granule version number (Auto-incrementing, nn in filenaming convention). Character max length of 20.
gran_segment	orbit segment of the granule (if more that 1 segment, use 0).Character max length of 1.

Files with INPUT_FILE and OUTPUT_FILE keywords must be listed in chronological order based on start and stop times. The start time of one file may overlap the stop time of another. In this case, data within the overlapping range will be written to the first file and not the second.

Example of a GLA file specification control line:

```
OUTPUT_FILE=gla01_002_11_0001_0043_4_00_0000.dat 85398 86000 1 1 0
OUTPUT_FILE=gla02_002_11_0001_0043_0_00_0000.dat 85398 86000 1 1 0
```

Example of an ANC file specification control line:

```
INPUT_FILE=anc07_001_01_0000.dat 85398 86000
INPUT_FILE=anc07_001_01_0001.dat 85398 86000
```

5.1.6 Example Control File

Following is a partial control file which highlights elements common to all GSAS control files. Examples of specific control files will be provided within the documentation for each specific PGE.

```
#----------Start of Control File
#
# <description of control file>
#
#----------Execution Information
#
=GLAS_L1A
TEMPLATE=<template name>
EXEC_KEY=<unique execution key>
DATE_GENERATED=<date/time control file was generated>
OPERATOR=<operator name/uid>
PGE_VERSION=<PGE version number>
#
#----------Static ANC Files
#
# Common Input ANC07 Files: 00=error, 01=global
#
INPUT_FILE=<anc0700_error_filename starttime stoptime>
INPUT_FILE=<anc0701_global_filename starttime stoptime>
#
#----------Output ANC06 File
#
# Output ANC06
#
OUTPUT_FILE=<anc06_filename starttime stoptime>
```

{processing scenario-specific elements}

```
#
#----------End of Control File
#
=End of Control File
```

5.2 Processing Controls

The core PGEs (GLAS_L1A, GLAS_Atm, GLAS_Alt) provide for flexible methods of processing GLAS data. All controls are specified via the control file. Methods of controlling data processing include:

- Input data time selection
- Output data time selection
- Execution control

The detailed documentation for each PGE will identify what respective types of processing controls are available.

5.2.1 Input Data Time Selection

As seen in the Control File section, all input files are required to be delimited by start and stop times on their control file entry. PGEs which support time selection will skip that data which are outside the limits defined by start and stop times. This data will be read, but not processed. Additionally, given the case of multiple input files of the same type, the PGE will seemlessly skip from one file to the next when all data from the current file have been read (or skipped via time selection).

Certain input ancillary files do not support input time selection but require, none the less, start and stop times in their control file entry. The recommended method of delimiting time in the case is to use the minimum-to-maximum range of input dynamic data time for these entries.

5.2.2 Output Data Time Selection

As with input files, all output files are required to be delimited by start and stop times on their control file entry. PGEs which support time selection will not write that data which are outside the limits defined by start and stop times. Additionally, given the case of multiple output files of the same type, the PGE will seemlessly skip from one file to the next when the current data time falls outside the range of the current output file. It is important to note that input data time selection and output data time selection are completely independent of one another. There is, however, a practical relationship between the two, since output data for a particular time cannot be written if no input data for that time is read (or specified).

5.2.3 Execution Scenarios

Most core PGEs permit multiple execution scenarios. Certain sets of computations have been grouped together by the software designers. Execution of these sets can be specified via specific execution flags with the PGE control file. The detailed documentation for each PGE specifies what execution flags are available and the processes they control. Additionally, there are dependencies between input file type, output file type, and the execution flags. These dependencies define execution scenarios, which will be described in the respective PGE -specific section of this document.

5.3 ANC07 Constants Files

ANC07 files are used to provide GSAS with change-controlled parameters provided by the Science Team and used during processing of GLAS data. These parameters were carefully selected such that the values could be modified without forcing a recompilation of the processing software. It is critical that these files are tightly change-controlled since unapproved modification could result in erroneous data being generated during the creation of the GLAS Products.

There are several types of ANC07 files. These types include a global constants file, an error file, and constants files specific to each of the science algorithm categories. The error file is detailed in the error section of this document. The science-algorithm specific constants files will be detailed in the respective documentation for those particular executables. This section will document the global constants file and the commonalities of all these files.

5.3.1 Use

The constants files are specified as input files within a particular executable's control file. The global constants file and the error constants file are required for all executables. Table 5-6 shows the naming convention of each ANC07 file and a description of its content. (*mmm* is the release number for process that created the file and is the version number of the file itself within the *mmm* release).

Table 5-6 ANC07 files

Naming Convention	Content
anc07_mmm_nn_0000.dat	Error/Status constants
anc07_mmm_nn_0001.dat	Global constants
anc07_mmm_nn_0002.dat	Atmosphere constants
anc07_mmm_nn_0003.dat	Elevation constants
anc07_mmm_nn_0004.dat	Waveform constants
anc07_mmm_nn_0005.dat	L1A constants
anc07_mmm_nn_0006.dat	Utility constants

5.3.2 Format

GSAS ANC07 files are delimited by section identifiers which differ (by design) from control files section identifiers. Each section is bounded by the section name and an "=". The section delimiters are defined as follows:

BEG_OF_STATUS=
...Status section contents...
END_OF_STATUS=

BEG_OF_ERROR=
...Error section contents...
END_OF_ERROR

BEG_OF_GLOBALS=
...Global constants section contents...
END_OF_GLOBALS

BEG_OF_ATM=
...Atmosphere constants section contents...
END_OF_ATM

BEG_OF_ELEV=
...Elevation constants section contents...
END_OF_ELEV

BEG_OF_L1A=
...L1A constants section contents...
END_OF_L1A

All GSAS ANC07 files are created in standard GSAS “keyword=value” format. This format is text-based and consists of a line containing a keyword/value pair delimited by an equal sign (=). The ordering of the keywords is not relevant but should follow a convention for consistency. Multiple instances of keywords are not allowed. The keyword is not case sensitive. Spaces are allowed, but not required. Comment lines must be prepended by a “#” character. The keyword and value are limited to 255 characters each.

5.3.3 ANC07 Output Options

GSAS provides the ability to write the values of selected sets of constants into the ANC06 file at runtime. This is activated by a control file entry with a keyword of WRITE_CONST. The associated value determines what set of constants should be written. Multiple WRITE_CONST control file entries, each specifying a different set of constants, are permitted.

Table 5-7 WRITE_CONST Control Values

Value	Constants to Write
ERR	Error/Status constants.
L1A	L1A constants.
WF	Waveform constants.
ATM	Atmosphere constants.
ELEV	Elevation constants.
GLOB	Global constants.

Example control file entries:

```
#Dump constants to ANC06
WRITE_CONST=GLOB
WRITE_CONST=L1A
```

5.3.4 Global Constants File Content

The ANC07 global constants file contains very-high level constants that are be used by more than one PGE. A sample global constants file follows:

```
##
# GLAS ANC07 Constants File
#
# 06/07/99 PLJ Corrected the elevation status messages
# 1999 September 22 JLee, Updated GLAS_Error where error < 0
# 09/27/99 GMM Updated the error list to include entries from ATM, WF, ELEV
#              subsystems and made error codes negative.
# 1999 October   18 JLee, Fixed error code.
# 1999 November  01 JLee, Integrated ATM constants
```

```
# 2000 March     13 JLee, Updated
# 2001 April     13 JLee, Added DEM constants from Elevation
# 2001 May       16 PJester, Added GPS to UTC offset
# 2001 June      08 JLee, MOdified GPS to UTC offset to match Donghui's data
# 2001 August    22 JLee, Moved calibration coefs from L1A to here so that
#                         both WF and L1A can use them
#
# Constants Mod Entries
#
BEG_OF_GLOBALS = -------------------------------------------------------------
GLOB_VERS = ANC07 Globals v2.0 October 2001
#
# Mean Ellipsoid Radius
#
ELLIPAE = 6378136.49d0
#
# Inverse of Ellipsoid Flattening
#
ELLIPF = 298.25645d0
#
# delta lat for DEM grid
#
DEMdLat  =  0.00833333333333d0
#
# delta lon for DEM grid
#
DEMdLon =  0.00833333333333d0
#
# Start latitude on the DEM grid
#
DEMLAT_BEG  = 90.00416666666667d0
#
# Start longitude on the DEM grid
#
DEMLON_BEG =  -0.00416666666667d0
#
# The offset from the GPS reference time (00:00 Jan 1, 1980)
# to the UTC reference time (12:00 Jan 1, 2000)
#
GPStoUTCoffset =  -2001.10980499199991
#
# Max PAD offset allowed (max off-nadir pointing angle)
#
MAXPADOFF =  0.5d0
#
# 1064 energy Calibration coefficient for Received WFs
#
gd_calibcoefrec = 0.002351d0
#
# 1064 energy Calibration coefficient for transmitted  WFs
#
gd_calibcoeftr = 0.00018d0
#
END_OF_GLOBALS = -------------------------------------------------------------
#
END_OF_ANC07_FILE = ----------------------------------------------------------
```

5.4 Invalid Values and Error/Status Reporting

This section documents the use of standardized methods of dealing with invalid data and error/status conditions.

5.4.1 Invalid Values

Not all data received from GLAS will be suitable for science processing. In addition, given the nature of the raw telemetry packets, some data may be missing. The concept of an "invalid value" is used to signify that data is invalid or missing and should not be used for processing. Invalid values are datatype-specific values which are defined in the GLAS global constants module. These variables are assigned to Product variables in order to indicate invalid or missing data. These values are defined in Table 5-8.

Table 5-8 Invalid Values

Datatype	Invalid Value
1 byte integer	127
2 byte integer	32767
4 byte integer	2147483647
4 byte real	3.40282E+38 x7F7FFFFF
8 byte real	1.797693094862316E+308 x7FEFFFFFFFFFFFFF

5.4.2 Error and Status Codes

GSAS uses a common error/status reporting facility. This ensures that error/status reporting is handled in a consistent manner throughout the software. This facility is based on the ANC07 error file and is configurable by the user.

5.4.2.1 ANC07 Error File Format

The ANC07 error file is in standard GSAS "keyword=value" format. See Section 5.3.2 for details of the format. As with other ANC07 files, the sections for error and status must be delimited by section identifiers. Identifiers for each section are listed below.

```
BEG_OF_STATUS=
...Status section contents...
END_OF_STATUS=

BEG_OF_ERROR=
...Error section contents...
END_OF_ERROR
```

The format of the error/status content is defined in Figure 5-1. The keyword can have

```
KEYWORD=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttxsxffffff
```

Figure 5-1 Error Ancillary File Format

the value of "ERROR" or "STATUS". This identifies if the line contains an error or status entry. The value is a text string with the specific format defined in Table 5-9.

Table 5-9 Error String Format

Character	Positions	Description
n	1-6	Error code (must be sequential within a section)
x	7,58,60	Space character (delimiter)
t	8-57	Message
s	59	Error severity (see Table 5-11)
f	61-66	Frequency of reporting (message is reported on 1st occurrence, then every f'th time)

There is a specific error number for each error/status value. Within the ANC07 file, these error numbers are numerically split into multiple sub-sections. Errors have negative numeric designations; status messages have positive designations.

Each major portion of the GSAS software supported by the specific error file begins at a different subsection number. Within a subsection, error numbers must be consecutive The use of sub-sectioning is optional for a simple error file. The GSAS ANC07 error file has 5 subsections. Table 5-10 lists each of the subsections and their starting error/status number.

Table 5-10 Error Sections

Starting Numbers	Description
-10001/10001	General error/status.
-20001/20001	L1A error/status.
-30001/30001	Waveform error/status
-40001/40001	Atmosphere error/status
-50001/50001	Elevation error/status.

5.4.2.2 Use

GLAS error messages are designed to inform a user when the software has encountered a problem. GLAS status messages are designed to assist the user in observing the flow of the processing. Status messages usually alert the user when the software begins execution of a subroutine. A great deal of flexibility was designed into this software in order to allow the user to customize the error/status display.

The user may modify error and status entries in order to configure the severity of the error and frequency of printout. The user is cautioned to seek GLAS change-control board approval before modifying the severity of an error. GSAS software will terminate processing upon receipt of a fatal severity code. Thus, modifying the severity may enable the software to execute in a non-tested mode.

The severity column controls how the GLAS software reacts when an error occurs. The 4 levels of severity are described in Table 5-11. GLAS software will terminate on a Fatal error. The frequency column controls how often an error message is printed out. The first instance of a specific error is always printed. Subsequent instances are printed out at the frequency specified. All instances are counted and the number of occurrences printed in an output summary.

Table 5-11 Error Severity Codes

Severity	Description
0	No error
1	Information/status
2	Warning
3	Fatal

5.4.2.3 Error/Status File Content

GLAS Error/Status messages and explanatory text are found in Appendix D of this document. HP Runtime error listings are found in Appendix D, as well.

5.4.2.4 Sample ANC07 Error/Status File

Following is a sample ANC07 error/status file. It has been trimmed down significantly for brevity.

```
#
# GLAS ANC07 Error File
#
# STATUS Section
#
BEG_OF_STATUS= --------------------------------------------------------------
STAT_VERS = ANC07 Status VERSION #
#
# Exec
#
#TATUS=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
STATUS=10001  Number of Records Processed:                       0     10
STATUS=10002  Number of Records Read:                            0     10
#
# L1A
#
#TATUS=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
STATUS=20001  L_Alt subroutine status                            0     10
```

```
STATUS=20002  L_Atm subroutine status                          0     10
#
# Waveform
#
#TATUS=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
STATUS=30001  W_Assess subroutine status                       0   9000
STATUS=30002  W_FunctionalFt subroutine status                 0   9000
#
# Atmosphere
#
#TATUS=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
STATUS=40001  A_aer_lays subroutine status                     0   7000
STATUS=40002  A_aer_opt_prop subroutine status                 0   7000
#
# Elevation
#
#TATUS=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
STATUS=50001  In E_CalcLoadTd subroutine status                0   1000
STATUS=50002  IN E_CalcOceanTd subroutine status               0   1000
#
END_OF_STATUS = --------------------------------------------------------------
#
#============================================================================
#
# ERROR Section
#
BEG_OF_ERROR = ---------------------------------------------------------------
ERR_VERS = ANC07 Error V2_exec_split 12 February 2001
#
# Exec
#
#RROR=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
ERROR=-10001 Error Opening File for Input:                    3      1
ERROR=-10002 No control file specified.                       3      1
#
#  L1A
#
#RROR=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
ERROR=-20001 Error reading PAD data Eng data                  3      2
ERROR=-20002 Error reading PAD data Eng data                  3      1
#
# Waveform
#
#RROR=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
ERROR=-30001 Singular Matrix                                  1      1
ERROR=-30002 Number of peaks found greater than max           1   9000
#
# Atmosphere
#
#RROR=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
ERROR=-40001 DEM out-of-bounds                                1     50
ERROR=-40002 532 integrated return flag poor or bad           1     50
#
# Elevation
#
#RROR=nnnnnnxttttttttttttttttttttttttttttttttttttttttttttttttttttxnxnnnnnn
ERROR=-50001  Error reading met height file                   3      1
ERROR=-50002  Error reading met temperature file              3      1
#
```

```
END_OF_ERROR = ------------------------------------------------------------
#
#
END_OF_ANC07_FILE = -------------------------------------------------------
```

5.4.3 Result Codes

All GSAS PGEs return a result code indicating success or failure. The result code may be checked by referencing $? (bourne sh). The result code values are a subset of the error/status codes and are defined in Table 5-12.

Table 5-12 Result Codes

Severity	Description
0	No error
3	Fatal error

5.5 ANC06 Metadata/Log File

5.5.1 ANC06 Overview

GSAS software creates ANC06 output files as required by GSDP-31100 (refer to the GLAS Science Software Requirements Document). The ANC06 file contains processing information, error messages, and status messages.

5.5.2 Format

The file is in a modified version of the GSAS keyword=value format. The format of an ANC06 entry is:

```
[rec_ndx] [keyword]=[value]
```

The first field [rec_ndx] is the current record index. This correlations to the time of the data being processed when the entry was written (if no data have been processed, the value may be 0 or an invalid value). The second field [keyword] is a keyword describing the type of information presented. The third field [value] is a formatted text message describing the event. Comments are allowed in order to group messages logically. Comment lines are pre-pended by the pound (#) sign.

The value field has several subfields. The first field is the numeric error code. The second field is the error severity. The third field is the name of the routine which reported the error. The fourth field is the standard error text with optional detailed text. The format of the subfields within the value field is shown below:

```
error_num, severity, calling_routine, std_message opt_text
```

5.5.3 Use

ANC06 files are designed to be human-readable. However, there are techniques which can make retrieval of information from these files easier. The UNIX grep command is a valuable tool which can be used to extract information from the ANC06 files.

For example, to find all errors within an ANC06 file:

```
grep -i ERROR anc06_filename
```

To find all version information,

```
grep -i VERS anc06_filename
```

To display control file information:

```
grep -i CONTROL anc06_filename
```

5.5.4 ANC06 Example and Description

An example ANC06 file is provided in Appendix C.

Section 6

GLAS_L0proc

6.1 Function

GLAS_L0proc (sometimes referred to as GLOP) reads GLAS Level-0 APID files, ANC47 construction records, ANC07 constants, and the ANC33 time conversion file to create ANC29 and ANC32 ancillary files which will be used in L1A processing. The ANC29 file is used by GLAS_L1A to read the APID files in a time-aligned manner. The ANC32 file is used by GLAS_L1A to allow the computation of precise timing information via a GPS reference. GLAS_L0proc should be run once all GLAS APID files for a particular 6-hour period have been received by the data processing facility. GLAS_L0proc does not permit multi-granule input or output files.

6.2 Input Files

Table 6-1 lists the required inputs to GLAS_L0proc. Files which are specific to GLAS_L0proc are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the those files not specific to GLAS_L0proc.

Table 6-1 GLAS_L0proc Inputs

File Spec	Type	Source	Short Description
gla00*_????.dat	Level-0 APID	EDOS	GLAS Level-0 APID files (one file per each APID type).
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error file.
anc07*_0001.dat	Static Ancillary	Science Team	GLAS global constants file.
anc47*_????.dat	Dynamic Ancillary	EDOS	APID construction records.
anc33*.dat	Dynamic Ancillary	ISIPS Operations	Counter-to-UTC conversion file.
Control File	Control	ISIPS Operations	Control file.

6.2.1 GLA00 APID Files

The GLA00 APIDs are Level-0 multi-rate spacecraft data files provided to the GLAS data processing facility by EDOS. There is a separate file for each specific APID type received from the spacecraft. These files are documented by the GLAS Instrument Team.

After delivery from EDOS, these files are renamed to comply with GLAS standard naming conventions by SDMS. The start time in the control file entry for the APIDs is used to determine the approximate UTC time for the data. This is important for finding the correct MET-to-UTC conversion entry in the ANC32 file.

GLAS_L0proc processes the APIDs listed in Table 6-2.

Table 6-2 APIDs Processed by GLAS_L0proc

APID	Filename	Description
12	gla00*_0012.dat	Altimeter Digitizer Large Sci Pkt
13	gla00*_0013.da	Altimeter Digitizer Small Sci Pkt
14	gla00*_0014.dat	Altimeter Digitizer Eng Pkt
15	gla00*_0015.dat	Photon Counter Sci Pkt
16	gla00*_0016.dat	Photon Counter Eng Pkt
17	gla00*_0017.dat	Cloud Digitizer Sci Pkt
18	gla00*_0018.dat	Cloud Digitizer Eng Pkt
19	gla00*_0019.dat	Ancillary Science Pkt
20	gla00*_0020.dat	CT HW telemetry #1 Data Pkt
21	gla00*_0021.dat	CT HW Telemetry #2 Data Pkt
22	gla00*_0022dat	CT HW Telemetry #3 Data Pkt
23	gla00*_0023.dat	CT HW telemetry #4 Data Pkt
24	gla00*_0024.dat	Small Software #1 Tlm
25	gla00*_0025.dat	Large Software Telemetry #1 Packet
26	gla00*_0026.dat	LPA Data Pkt
27	gla00*_0027.dat	Memory Dwell Packets 1
28	gla00*_0028.dat	Memory Dwell Packets 2
31	gla00*_0031.dat	DSP Code Memory Dump
32	gla00*_0032.dat	DSP Data Memory Dump
33	gla00*_0033.dat	C & T Dwell Packet
34	gla00*_0034.dat	Event Message Packet
35	gla00*_0035.dat	Memory Dump Packet
36	gla00*_0036.dat	Table Dump Packet
38	gla00*_0038.dat	Boresite Calibration Packet
48	gla00*_0048.dat	GLAS Data Types Packet
49	gla00*_0049.dat	Command History Packet
50	gla00*_0050.dat	CT HW telemetry #5 Data Pkt
55	gla00*_0055.dat	Large Software Telemetry #2 Packet
126	gla00*_0126.dat	LPA Test Packet.
1984	gla00*_1984.dat	GLAS PRAP Packet

6.2.2 ANC33 Counter to UTC Conversion File

The ANC33 file is used to convert mission-elapsed time (MET), which is provided in the APIDs, to GLAS-standard UTC time. Since the MET can be re-set by a roll-over or a spacecraft upset it is important that this file be maintained and provided to the GLAS processing facility in a timely manner. The file is delivered to ISIPS from the ISF as described in the ISF/ISIPS Interface Control Document.

ANC33 file is a ANSI text file. Each line contains data for a single entry in the file (data should not be hard wrapped). Comment lines are allowed and prepended by a # character. Each line contains the following information:

```
d_shdr_count <sp> d_shdr_count_prap <sp> d_utc <sp>
d_glas_osc_rate <sp> d_sc_osc_rate <sp> d_tdelay_digtzr <sp>
d_rdelay_digtzr <sp> d_plTbias <sp> d_plRbias <sp>
i_trkr_subject1 <sp> i_trkr_subject2 <sp> i_trkr_subject2
<sp>implement_time
```

Each field is defined in Table 6-3.

Table 6-3 ANC33 Field Descriptions

Field	Description
d_shdr_count	double precision : the counter value in the secondary header on MOST APIDS
d_shdr_count_prap	double precision: the counter value in the secondary header of PRAP
d_utc	double precision: the J2000 UTC time in seconds to which the counter values are converted
d_glas_osc_rate	double precision: the GLAS oscillator rate
d_sc_osc_rate	double precision: the Spacecraft oscillator rate
d_tdelay_digtzr	double precision: time delay for digitizer in seconds
d_rdelay_digtzr	double precision: internal range delay for digitizer in m
d_plTbias	double precision: post launch time bias in seconds
d_plRbias	double precision: post launch range bias in m
i_trkr_subject(1)	integer: the subject indicator for LRS tracker 0
i_trkr_subject(2)	integer: the subject indicator for LRS tracker 1
i_trkr_subject(3)	integer: the subject indicator for LRS tracker 2
d_implement_time	double precision: the J2000 UTC time in seconds where the data for this entry are first valid

Note that the Implement_time is the UTC time at which this conversion was valid. GLAS_L0proc uses the designated start time of the first APID specified in the control file to find the correct position within the ANC33 file based on the Implement_time field.

Example:

```
#
# GLAS ISF/GLAS ISIPS Interface file
#
# This file is delivered to ISIPS from the ISF as described in the
# ISF/ISIPS Interface Control Document.
#
0000000000000000 0000000000000000 -00043200.0 1.0 1.0 0.00001511   9.5 0.0 0.0 0  1
2 0000000000000
0000000100000000 0000000200000000 -00043200.0 1.0 0.99999 0.00001501   9.6 0.0 0.1 0
2  1 0000000086400
```

Note that the data are read free-format and are not restricted to a certain precision or number of columns. Although the example lines are wrapped to fit on a page, each entry should be contained on a single line (soft wrapping is OK).

6.2.3 ANC47 Construction Record

EDOS provides construction records for each APID they provide to GSAS. Construction records have information which indicates if APID packets were "filled" by EDOS. A filled packet is one which was corrupt or incomplete when received from a ground station. GLAS_L0proc parses the construction record, determines the position of filled packets, and then skips those packets during processing.

6.2.4 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_L0proc are described in this section.

The control file section delimiter for GLAS_L0proc is:

```
=GLAS_L0P
```

Since GLAS_L0proc has no requirement for execution scenarios, there are no unique keywords for the GLAS_L0proc control file. GLAS_L0proc will perform all functions based on the presence of input and output files within the control file.

6.3 Output Files

Table 6-4 lists the outputs created by GLAS_L0proc. Files which are specific to GLAS_L0proc are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the files which are not specific to GLAS_L0proc.

6.3.1 ANC29 Index File

The ANC29 index file provides GLAS_L1A with a method of time-correlating the GLAS APID files. It contains an index record for every record in the set of input APID files. ANC29 is a binary, fixed-length record file. Its content is described in Table 6-5.

Table 6-4 GLAS_L0proc Outputs

File Spec	Type	Destination	Short Description
anc29*.dat	Dynamic Ancillary	GLAS_L1A	Index file correlating APID times.
anc32*.dat	Dynamic Ancillary	GLAS_L1A	Frequency board to GPS time correlation file.
anc06*.dat	Dynamic Ancillary	ISIPS Operations	Standard metadata/processing log file.

Table 6-5 ANC29 File Content

Type	Name	Description
double	utctime	J2000 UTC time in seconds of the APID.
long int	rec_ndx	Unique record index of the APID group. (Nominally, (utctime-offset) * 10)
long int	shot_ctr	Shot counter (if available).
long int	rec_num	Physical record number in the APID file.
long int	apid	APID identification number.
long int	dqflag	Data quality flag (0=good data).
short int	sort_order	Sort order (for internal use).
shor t int	spare	Spare (for internal use).

6.3.2 ANC32 GPS File

The ANC32 GPS file provides GLAS_L1A with a method of computing precise timing calculations based on the last update of the onboard GPS. It contains record which identified each time the GPS clock is updated within the APID packets. ANC32 is a binary, fixed-length record file and nominally occurs at a rate of one record every 10 seconds. Its content is described in Table 6-6.

Table 6-6 ANC32 File Content

Type	Name	Description
long int	rec_ndx	Unique record index of the APID group. (Nominally, (utctime-offset) * 10)
short int	i_ScPosPktShot	APID 19/1182 Shot count for 1553 Spacecraft Time & Pos packet.
short int	i_useflag	Flag indicating if the data are valid (0=valid, other=not valid)
double	d_utctime	J2000 UTC time in seconds from secondary header of packet where GPS changed.
double	d_FTLatch	APID 19/1195 GPS 10 Sec Pulse 40 bit count value

Table 6-6 ANC32 File Content

Type	Name	Description
double	d_ScPosPktGMET	APID 19/1184 GLAS MET for 1553 Spacecraft Time & Pos packet.
double	d_VTCW	APID 19/1142 Spacecraft VTCW (start of SC Time & Pos pkt)
double	d_VTCWp	APID 19/1182 VTCW and at time of 0.1Hz pulse (end of SC Time & Pos pkt)
double	d_GPSTime	APID 19/1172 GPS time in Seconds
double	d_GPSppsGMET	PID 19/1201 GLAS MET for GPS 0.1 Hz Pulse

6.4 Processing and Reprocessing Scenarios

GLAS_L0proc supports only a full-processing scenario. Additionally, all available APIDs files and ANC47 construction records for a specific time period must be processed during the same run. Only one valid ANC29/32 file is permitted for a time period. If additional corresponding APIDs are received at a later date, <u>all</u> of the APIDs for that particular period must be re-processed to create replacement ANC29/32 files.

Time-selected processing is not supported. The start and stop times on the INPUT_FILE and OUTPUT_FILE control file entries are ignored for processing, <u>but are critical for finding the correct time of data during ANC33 file searche</u>s.

6.5 Startup and Termination

This is an overview of the steps necessary to run GLAS_L0proc.

6.5.1 Setup a Runtime Directory

The suggested method of running GLAS_L0proc is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a GLAS_L0proc run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_L0proc .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc07*.dat .
ln -s $GLAS_HOME/data/anc33*.dat .
ln -s $DATA_DIR/*.dat .
```

6.5.2 Create a Control File

GLAS_L0proc is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

```
cp $GLAS_HOME/data/cf_glas_l0proc.ctl ./control_file_name
vi ./control_file_name
```

6.5.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

```
./GLAS_L0proc control_file_name
```

6.5.4 Run-Time Statistics

GLAS_L0proc processes 6 hours of GLAS data in less than 2 minutes. This may vary due to hardware load and data dependencies.

6.5.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

6.6 Error and Warning Messages

All GSAS error and warning messages are listed numerically in Appendix D.

6.7 Recovery Steps

If GLAS_L0proc terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem. Refer to errors listed in Appendix D for assistance in determining the problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_L0proc will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions.

6.8 Sample GLAS_L0proc Control File

```
=GLAS_L0P
#
#----------Start of Control File
#
# This is an L0proc control file
```

```
#
#---------Execution Information
#
EXEC_KEY=p1t9
DATE_GENERATED=26 June 2002
OPERATOR=SBhardwaj
#
#----------Input GLA Files
#
# Input ANC07
#
INPUT_FILE=anc07_001_01_0000.dat 0 100000000
INPUT_FILE=anc07_001_01_0001.dat 0 100000000
#
# Input GLA00 Files
#
INPUT_FILE=gla00_008_20020406_000000_01_0012.dat 0 100000000
INPUT_FILE=gla00_008_20020406_000000_01_0013.dat 0 100000000
INPUT_FILE=gla00_008_20020406_000000_01_0015.dat 0 100000000
INPUT_FILE=gla00_008_20020406_000000_01_0017.dat 0 100000000
INPUT_FILE=gla00_008_20020406_000000_01_0019.dat 0 100000000
INPUT_FILE=gla00_008_20020406_000000_01_1984.dat 0 100000000
INPUT_FILE=gla00_008_20020406_010000_01_0020.dat 0 100000000
INPUT_FILE=gla00_008_20020406_010000_01_0021.dat 0 100000000
INPUT_FILE=gla00_008_20020406_010000_01_0022.dat 0 100000000
INPUT_FILE=gla00_008_20020406_010000_01_0023.dat 0 100000000
INPUT_FILE=gla00_008_20020406_010000_01_0025.dat 0 100000000
INPUT_FILE=gla00_008_20020406_000000_01_0126.dat 0 100000000
INPUT_FILE=gla00_008_20020406_010000_01_0050.dat 0 100000000
INPUT_FILE=gla00_008_20020406_010000_01_0055.dat 0 100000000
#
# Input anc47 Files
#
INPUT_FILE=anc47_008_20020406_000000_01_0012.dat 0 100000000
INPUT_FILE=anc47_008_20020406_000000_01_0013.dat 0 100000000
INPUT_FILE=anc47_008_20020406_000000_01_0015.dat 0 100000000
INPUT_FILE=anc47_008_20020406_000000_01_0017.dat 0 100000000
INPUT_FILE=anc47_008_20020406_000000_01_0019.dat 0 100000000
INPUT_FILE=anc47_008_20020406_000000_01_1984.dat 0 100000000
INPUT_FILE=anc47_008_20020406_010000_01_0020.dat 0 100000000
INPUT_FILE=anc47_008_20020406_010000_01_0021.dat 0 100000000
INPUT_FILE=anc47_008_20020406_010000_01_0022.dat 0 100000000
INPUT_FILE=anc47_008_20020406_010000_01_0023.dat 0 100000000
INPUT_FILE=anc47_008_20020406_010000_01_0025.dat 0 100000000
INPUT_FILE=anc47_008_20020406_000000_01_0126.dat 0 100000000
INPUT_FILE=anc47_008_20020406_010000_01_0050.dat 0 100000000
INPUT_FILE=anc47_008_20020406_010000_01_0055.dat 0 100000000
#
# Input of time conversion files
#
INPUT_FILE=anc33_00_20020406_000000_00_0000.dat 0 100000000
#
#----------Output GLA Files
#
OUTPUT_FILE=anc06_20010823_180000_000234_glas_l0p.txt 0 100000000
OUTPUT_FILE=anc29_001_20010823_180000_01_0000.dat 0 100000000
OUTPUT_FILE=anc32_001_20010823_180000_01_0000.dat 0 100000000
#
#----------End of GLAS_L0proc Section---------------------------
```

Section 7

GLAS_L1A

7.1 Function

GLAS_L1A uses calibration and conversion equations determined during GLAS system testing to convert the measured counts within the GLA00 data into engineering units. Some data do not require conversion and will be copied directly from the input to the appropriate output product. GLAS_L1A computes precise time-tagging information for each laser shot and produces quality assurance measurements used to determine the quality of the data processed.

GLAS_L1A reads GLAS Level-0 APID files, ANC20, ANC25, ANC29, ANC32, ANC33, ANC45 and ANC07 ancillary files to create GLAS L1A product files. The ANC29 file is used by GLAS_L1A to read the APID files in a time-aligned manner. The ANC32 file is used by GLAS_L1A to allow the computation of precise timing information via a GPS reference. The ANC20 file is used to compute location from the predicted orbit. The ANC07 files are read to initialize constants and error/status messages. The ANC45 files are used to initialize metadata/header information.

ANC29, ANC32, and the corresponding GLA00 APIDs should be considered a single, inseparable data set. It is critical that the appropriate ANC29/32 files be provided for each distinctive set of GLA00 APIDs provided as input. ANC29 is the index by which the APIDs are read in a time-aligned fashion. Without ANC29 files which correlated with the input GLA00 APIDS, processing will fail. Likewise, without ANC32 files, precision time computations will fail.

GLAS_L1A should be run after successful execution of GLAS_L0proc. GLAS_L1A permits multi-granule input or output files, input/output time selection and selected execution.

7.2 Input Files

Table 7-1 lists the inputs files to GLAS_L1A. Files which are specific to GLAS_L1A are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details.

Table 7-1 GLAS_L1A Inputs

File Spec	Type	Source	Short Description
gla00*_????.dat	Level-0 APID	EDOS	Level-0 APID files (one file per each APID type).
anc07*_0000.dat	Static Ancillary	Science Team	Error file.
anc07*_0001.dat	Static Ancillary	Science Team	Global constants file.
anc07*_0005.dat	Static Ancillary	Science Team	L1A constants file.

Table 7-1 GLAS_L1A Inputs (Continued)

File Spec	Type	Source	Short Description
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc29*.dat	Dynamic Ancillary	GLAS_L0proc	Index file correlating APID times.
anc32*.dat	Dynamic Ancillary	GLAS_L0proc	Frequency board to GPS time correlation file.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.
anc20*.dat	Dynamic Ancillary	UTexas	Predicted orbit file.
anc45*_0001.dat	Static Ancillary	Science Team	GLA01 metadata input file.
anc45*_0002.dat	Static Ancillary	Science Team	GLA02 metadata input file.
anc45*_0003.dat	Static Ancillary	Science Team	GLA03 metadata input file.
anc45*_0004.dat	Static Ancillary	Science Team	GLA04 metadata input file.
Control File	Control	ISIPS Operations	Control file.

7.2.1 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_L1A are described in this section.

The control file section delimiter for GLAS_L1A is:

```
=GLAS_L1A
```

In order to satisfy the partial execution requirement, GLAS_L1A can perform limited processing based on execution flags within the control file. The flags and the processes they control are defined in Table 7-2.

Table 7-2 GLAS_L1A Control Flags

Flag	Significance
L1A_PROCESS=None	Indicates that GLAS_L1A should perform no processing.
L1A_PROCESS=L_ALT	Indicates that GLAS_L1A should perform altimetry-related processing.
L1A_PROCESS=L_ATM	Indicates that GLAS_L1A should perform atmosphere-related processing.
L1A_PROCESS=L_ENG	Indicates that GLAS_L1A should perform engineering processing.
L1A_PROCESS=L_ATT	Indicates that GLAS_L1A should perform SRS/GPS/laser pointing processing.
L1A_PROCESS=ALL	Indicates that GLAS_L1A should perform all processing.

Combinations of the execution flags are allowed. For example, to perform both altimetry and engineering processing, the control file would have the following entries:

```
L1A_PROCESS=L_ALT
L1A_PROCESS=L_ENG
```

A control file option is available to designate that location information be taken from the APID data rather than the ANC20 predicted orbit file. This is useful for quick-look processing when the ANC20 is not yet available but should not be used for normal, production processing. To indicate that pass-thru processing should be used, add the following line to the control file:

```
POSITION=PASS_THRU
```

If this line is not included in the control file, ANC20 processing will occur by default.

7.3 Output Files

Table 7-3 lists the output files created by GLAS_L1A. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the output files

Granule boundaries are designated by specifying start and stop times on the OUTPUT_FILE control file entries.

Table 7-3 GLAS_L1A Outputs

File Spec	Type	Destination	Short Description
gla01*.dat	L1A Product	GLAS_L1A	GLAS L1A Altimetry product file. Contains the waveforms and the altimeter and timing data required to produce higher level range and elevation products.
gla02*.dat	L1A Product	GLAS_Atm	GLAS L1A Atmosphere product file. Contains the normalized backscatter, photon counter, cloud digitizer, timing, and location data required to produce the higher level atmosphere data products.
gla03*.dat	L1A Product	Archive	L1A Engineering product file. Contains the GLAS instrument's engineering and housekeeping data.
gla04*_0001.dat	L1A Products	UTEXAS	L1A LPA product file.
gla04*_0002.dat	L1A Products	UTEXAS	L1A LRS product file.
gla04*_0003.dat	L1A Products	UTEXAS	L1A GYRO product file.
gla04*_0004.dat	L1A Products	UTEXAS	L1A IST product file.

Table 7-3 GLAS_L1A Outputs

File Spec	Type	Destination	Short Description
gla04*_0005.dat	L1A Products	UTEXAS	L1A BST product file.
gla04*_0006.dat	L1A Products	UTEXAS	L1A SCPA product file.
gla01*.qap	L1A Quality	QA	L1A Altimetry quality file.
gla02*.qap	L1A Quality	QA	L1A Atmosphere quality file.
gla03*.qap	L1A Quality	QA	L1A Engineering quality file.
gla04*_0000.qap	L1A Quality	QA	L1A SRS/GPS/laser pointing quality file.
anc06*.txt	Dynamic Ancillary	ISIPS Operations	Standard metadata/processing log file.

Of importance is the relationship between the execution flags and the output products. If a particular execution flag is present in the control file, then the corresponding output file type must be present as well. Table shows the relationship between output products and execution flags.

Table 7-4 GLAS_L1A Execution Flags and Requisite Output Products

Execution Flag	Required Output Product
L1A_PROCESS=L_ALT	gla01*.dat, qap01*.dat
L1A_PROCESS=L_ATM	gla02*.dat, qap02*.dat
L1A_PROCESS=L_ENG	gla03*.dat, qap03*.dat
L1A_PROCESS=L_ATT	gla04*.dat, gla04*.qap

7.4 Execution Scenarios

GLAS_L1A supports multiple execution scenarios. Control files defining the requisite execution flags as well as input and output files are shown for each supported scenario. For brevity, only those control file entries unique to the particular scenario are show. Additionally, even though multiple files of the same time may be needed, a single instance is show.

Since the ANC29 and ANC32 files directly correspond to specific sets of APIDs, great care must be taken to ensure that the relative order of input files within the control file is correct. For example, the APID files which correspond to the first ANC29 file must be listed first in the control file relative to other APID files. However, this is not really as complex as it sounds since following the rule that input files are listed within the control file in time order guarantees this will occur.

All execution scenarios require the L_ENG processes to provide values for the L_ALT and L_ATM processes. It is possible to run these scenarios without creating a GLA03 file by not specifying GLA03 as an output in the control file. This is not recommended since data used will have not been quality-checked by post-processing QA processes.

Time-selected processing may be performed by specifying specific start and stop times on the INPUT_FILE and OUTPUT_FILE control file entries.

During selective processing, only specific GLA00 APID input files are required. Processing will still occur if some of the APIDs are missing, but for full and complete processing those listed are required. It is expected that not all APIDs are available at all times (especially engineering-related APIDs). Table 7-5 lists the execution flags and requisite GLA00 APID inputs.

Table 7-5 GLAS_L1A Input APIDs

Execution Flag	Input APIDs
L1A_PROCESS=L_ALT	12/13, 19, 21
L1A_PROCESS=L_ATM	12/13, 15, 17, 19
L1A_PROCESS=L_ENG	12/13, 15, 17, 19, 20, 21, 22, 23, 24, 25, 50, 55
L1A_PROCESS=L_ATT	12/13, 19, 26, 1984

7.4.1 Standard Processing

This scenario is used for normal L1A processing, creating GLA01, GLA02, GLA03 and GLA04 data from GLA00 APIDs and the requisite ANC files. In this scenario, all GLA00 APIDs are read. All input ANC files are required.

7.5 Startup and Termination

This is an overview of the steps necessary to run GLAS_L1A.

7.5.1 Setup a Runtime Directory

The suggested method of running GLAS_L1A is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a GLAS_L1A run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_L1A .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc*.dat .
ln -s $DATA_DIR/*.dat .
```

7.5.2 Create a Control File

GLAS_L1A is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

cp $GLAS_HOME/data/cf_glas_l1a.ctl ./*control_file_name*
vi ./*control_file_name*

7.5.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

./GLAS_L1A *control_file_name*

7.5.4 Run-Time Statistics

GLAS_L1A processes 25 hours of GLAS data in approximately 3 hours. This may vary due to hardware load and data dependencies.

7.5.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

7.6 Error and Warning Messages

All GSAS error and warning messages are numerically listed in Appendix D.

7.7 Recovery Steps

If GLAS_L1A terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem. Refer to errors listed in Appendix D for assistance in determining the problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_L1A will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions.

7.8 Sample GLAS_L1A Control File

```
=GLAS_L1A
PGE_VERSION=2.2
PASSID=1 22010052332 51860152 51865951 123.9248348 4771
PASSID=2 22010052333 51865951 51871750  99.7298577 4772
PASSID=3 22010052334 51871750 51877549  75.5345768 4773
PASSID=4 22010052335 51877549 51883348  51.3401797 4774
INPUT_FILE=gla00_008_20020406_000000_01_0012.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_000000_01_0013.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_000000_01_0015.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_000000_01_0017.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_000000_01_0019.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_000000_01_1984.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_010000_01_0020.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_010000_01_0021.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_010000_01_0022.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_010000_01_0023.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_010000_01_0025.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_000000_01_0126.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_010000_01_0050.dat 51861600 51883201
INPUT_FILE=gla00_008_20020406_010000_01_0055.dat 51861600 51883201
INPUT_FILE=anc07_001_01_0000.dat 0 0
INPUT_FILE=anc07_001_01_0001.dat 0 0
INPUT_FILE=anc07_001_01_0002.dat 0 0
INPUT_FILE=anc07_001_01_0003.dat 0 0
INPUT_FILE=anc07_001_01_0004.dat 0 0
INPUT_FILE=anc07_001_01_0006.dat 0 0
INPUT_FILE=anc07_001_01_0005.dat 0 0
INPUT_FILE=ANC20_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC20_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=anc25_00_20020406_000000_00_0000.dat 0 0
INPUT_FILE=anc29_001_20010823_180000_01_0000.dat 51861600 51883201
INPUT_FILE=anc32_001_20010823_180000_01_0000.dat 51861600 51883201
INPUT_FILE=anc33_00_20020406_000000_00_0000.dat 0 0
INPUT_FILE=anc45_001_01_0001.dat 0 0
INPUT_FILE=anc45_001_01_0002.dat 0 0
INPUT_FILE=anc45_001_01_0003.dat 0 0
INPUT_FILE=anc45_001_01_0004.dat 0 0
OUTPUT_FILE=GLA01_002_22_01005_2332_1_01_0000.DAT 51860959 51862240 2 1 1
OUTPUT_FILE=GLA01_002_22_01005_2332_2_01_0000.DAT 51862240 51863856 2 1 2
OUTPUT_FILE=GLA01_002_22_01005_2332_3_01_0000.DAT 51863856 51865142 2 1 3
OUTPUT_FILE=GLA01_002_22_01005_2332_4_01_0000.DAT 51865142 51866758 2 1 4
OUTPUT_FILE=GLA01_002_22_01005_2333_1_01_0000.DAT 51866758 51868039 2 1 1
OUTPUT_FILE=GLA01_002_22_01005_2333_2_01_0000.DAT 51868039 51869655 2 1 2
OUTPUT_FILE=GLA01_002_22_01005_2333_3_01_0000.DAT 51869655 51870941 2 1 3
OUTPUT_FILE=GLA01_002_22_01005_2333_4_01_0000.DAT 51870941 51872557 2 1 4
OUTPUT_FILE=GLA01_002_22_01005_2334_1_01_0000.DAT 51872557 51873837 2 1 1
OUTPUT_FILE=GLA01_002_22_01005_2334_2_01_0000.DAT 51873837 51875453 2 1 2
OUTPUT_FILE=GLA01_002_22_01005_2334_3_01_0000.DAT 51875453 51876740 2 1 3
OUTPUT_FILE=GLA01_002_22_01005_2334_4_01_0000.DAT 51876740 51878356 2 1 4
#
OUTPUT_FILE=GLA02_002_22_01005_2331_0_01_0000.DAT 51855160 51866758 2 1 0
OUTPUT_FILE=GLA02_002_22_01005_2333_0_01_0000.DAT 51866758 51878356 2 1 0
OUTPUT_FILE=GLA03_002_22_01005_2331_0_01_0000.DAT 51855160 51866758 2 1 0
OUTPUT_FILE=GLA03_002_22_01005_2333_0_01_0000.DAT 51866758 51878356 2 1 0
#
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0001.DAT 51855160 51866758 2 1 0
```

```
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0001.DAT 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0002.DAT 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0002.DAT 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0003.DAT 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0003.DAT 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0004.DAT 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0004.DAT 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0005.DAT 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0005.DAT 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0006.DAT 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0006.DAT 51866758 51878356 2 1 0
#
OUTPUT_FILE=GLA01_002_22_01005_2332_1_01_0000.DAT 51860959 51862240 2 1 1
OUTPUT_FILE=GLA01_002_22_01005_2332_2_01_0000.DAT 51862240 51863856 2 1 2
OUTPUT_FILE=GLA01_002_22_01005_2332_3_01_0000.QAP 51863856 51865142 2 1 3
OUTPUT_FILE=GLA01_002_22_01005_2332_4_01_0000.QAP 51865142 51866758 2 1 4
OUTPUT_FILE=GLA01_002_22_01005_2333_1_01_0000.QAP 51866758 51868039 2 1 1
OUTPUT_FILE=GLA01_002_22_01005_2333_2_01_0000.QAP 51868039 51869655 2 1 2
OUTPUT_FILE=GLA01_002_22_01005_2333_3_01_0000.QAP 51869655 51870941 2 1 3
OUTPUT_FILE=GLA01_002_22_01005_2333_4_01_0000.QAP 51870941 51872557 2 1 4
OUTPUT_FILE=GLA01_002_22_01005_2334_1_01_0000.QAP 51872557 51873837 2 1 1
OUTPUT_FILE=GLA01_002_22_01005_2334_2_01_0000.QAP 51873837 51875453 2 1 2
OUTPUT_FILE=GLA01_002_22_01005_2334_3_01_0000.QAP 51875453 51876740 2 1 3
OUTPUT_FILE=GLA01_002_22_01005_2334_4_01_0000.QAP 51876740 51878356 2 1 4
#
OUTPUT_FILE=GLA02_002_22_01005_2331_0_01_0000.QAP 51855160 51866758 2 1 0
OUTPUT_FILE=GLA02_002_22_01005_2333_0_01_0000.QAP 51866758 51878356 2 1 0
OUTPUT_FILE=GLA03_002_22_01005_2331_0_01_0000.QAP 51855160 51866758 2 1 0
OUTPUT_FILE=GLA03_002_22_01005_2333_0_01_0000.QAP 51866758 51878356 2 1 0
#
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0001.QAP 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0001.QAP 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0002.QAP 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0002.QAP 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0003.QAP 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0003.QAP 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0004.QAP 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0004.QAP 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0005.QAP 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0005.QAP 51866758 51878356 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0006.QAP 51855160 51866758 2 1 0
OUTPUT_FILE=GLA04_002_22_01005_2333_0_01_0006.QAP 51866758 51878356 2 1 0

OUTPUT_FILE=ANC06_001_20010823_180000_01_00_glas_l1a.TXT 51861600 51883201
L1A_PROCESS=ALL
```

Section 8

GLAS_Alt

8.1 Function

GLAS_Alt uses algorithms defined in the GLAS Altimetry ATBD to process the GLAS waveforms and compute elevation parameters. GLAS_Alt functionality is divided into waveform and elevation processes. The waveform processes compute waveform parameters which are stored in the GLA05 file. The elevation processes use the waveform parameters, and (optionally) L2 atmosphere parameters, to compute elevation measurements which are stored in the GLA06 file and on the region-specific GLA12-15 files.

GLAS_Alt reads GLAS Level-1A product files and requisite ancillary files to create GLAS Altimetry product files. In its most basic execution, GLAS_Alt reads the GLA01 products created by GLAS_L1A to create GLA05, GLA06, and GLA12-15 roducts. However, in some execution scenarios, GLAS_Alt can read the GLA05 or GLA06 files created in a previous execution to produce GLA06 and/or GLA12-15 products. GLAS_Alt uses the ANC07 files for constants and error/status messages. It uses the global Digital Elevation Model from ANC12 DEM files. It uses the ANC13 file to determine the geoid, ANC16/17 to calculate tides and ANC01 for meteorological information. It uses ANC09 for precision attitude, and ANC04/ANC08 for precision orbit information. It uses ANC27 for regional mask information to define what data are placed on the regional products (GLA12-15). It uses ANC45 files to initialize metadata/header information. ANC25 and ANC33 files are used for time delay and oscillator-based corrections.

GLAS_Alt should be run after successful execution of GLAS_L1A, met_util, and (optionally) GLAS_Atm. GLAS_Alt may be run all at once or in two separate executions: the first for waveform processing, the second for elevation processing. GLAS_Alt permits multi-granule input/output files, input/output time selection, selected execution, and reprocessing.

8.2 Input Files

Table 8-1 lists the inputs files to GLAS_Alt. Files which are specific to GLAS_Alt are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details.

8.2.1 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_Alt are described here.

The control file section delimiter for GLAS_Alt is:

```
=GLAS_Alt
```

Table 8-1 GLAS_Alt Inputs

File Spec	Type	Source	Short Description
anc01*.dat	Dynamic Ancillary	met_util	Meteorological subset files. Data sets at times before and after the time of the profile are interpolated to the time of the profile.
anc04*.dat	Dynamic Ancillary	UTexas	IERS Polar Motion and Earth Rotation Data File.
anc07*_0000.dat	Static Ancillary	Science Team	Error file.
anc07*_0001.dat	Static Ancillary	Science Team	Global constants file.
anc07*_0003.dat	Static Ancillary	Science Team	Waveform constants file. *Waveform only
anc07*_0004.dat	Static Ancillary	Science Team	Elevations constant file. *Elevation only
anc08*.dat	Dynamic Ancillary	UTexas	Precision Orbit file.
anc09*.dat	Dynamic Ancillary	UTexas	Precision Attitude file.
anc12*_0000.dat	Static Ancillary	Science Team	Coarse DEM file *Elevation only.
anc12*_0001.dat	Static Ancillary	Science Team	Fine DEM file *Elevation only.
anc13*.dat	Static Ancillary	Science Team	Geoid file *Elevation only.
anc16*.dat	Static Ancillary	Science Team	Load tide coefficients file *Elevation only.
anc17*.dat	Static Ancillary	Science Team	Ocean tide coefficients file *Elevation only.
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc27*_0000.dat	Static Ancillary	Science Team	Coarse regional mask file.
anc27*_0001.dat	Static Ancillary	Science Team	Fine regional mask file.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.
anc41*.dat	Dynamic Ancillary	Science Team	JPL Planetary Ephemeris
anc45*_0001.dat	Static Ancillary	Science Team	GLA01 metadata input file. *Waveform only
anc45*_0005.dat	Static Ancillary	Science Team	GLA05 metadata input file.
anc45*_0006.dat	Static Ancillary	Science Team	GLA06 metadata input file.
anc45*_0012.dat	Static Ancillary	Science Team	GLA12 metadata input file.

Table 8-1 GLAS_Alt Inputs (Continued)

File Spec	Type	Source	Short Description
anc45*_0013.dat	Static Ancillary	Science Team	GLA13 metadata input file.
anc45*_0014.dat	Static Ancillary	Science Team	GLA14 metadata input file.
anc45*_0015.dat	Static Ancillary	Science Team	GLA15 metadata input file.
Control File	Control	ISIPS Operations	Control file.
gla01*_.dat	Level-1A Product	GLAS_L1A	L1A Altimetry product file. *Waveform only.
gla05*_.dat	Level-1B Product	GLAS_Alt	L1B Waveform product file. *Elevation only
gla09*_.dat	Level-2 Product	GLAS_Atm	L2 Atmosphere product file. *Optional-Elevation only
gla11*_.dat	Level-2 Product	GLAS_Atm	L2 Atmosphere product file. *Optional-Elevation only
gla06*_.dat	Level-1B Product	GLAS_Alt	L1A Elevation product file. *Elevation only

In order to satisfy the partial execution requirement, GLAS_Alt can perform limited processing based on execution flags within the control file. The flags and the processes they control are defined in Table 8-2.

Table 8-2 GLAS_Alt Control Flags

Flag	Significance
WAVEFORM_PROCESS=None	Indicates that GLAS_Alt should perform no waveform processing.
WAVEFORM_PROCESS=All	Indicates that GLAS_Alt should perform waveform processing.
WAVEFORM_OPTIONS=w_noLandCalc	Indicates that GLAS_Alt should not process data using alternate parameters
WAVEFORM_OPTIONS=w_noOtherCalc	Indicates that GLAS_Alt should not process data using standard parameters
WAVEFORM_OPTIONS=w_noGeo	Indicates that GLAS_Alt should not compute latitude, longitude, or elevation, but, instead, will use pass-thru data from an input GLA05. This flag should only be used in a reprocessing scenario where a GLA05 in input as well as output.
ELEVATION_PROCESS=None	Indicates that GLAS_Alt should perform no elevation processing.
ELEVATION_PROCESS=E_CalcLoadTD	Indicates the Load Tide computations should be performed.

Table 8-2 GLAS_Alt Control Flags (Continued)

Flag	Significance
ELEVATION_PROCESS=E_CalcOceanTD	Indicates the Ocean Tide computations should be performed.
ELEVATION_PROCESS=E_CalcEarthTD	Indicates the Earth Tide computations should be performed.
ELEVATION_PROCESS=E_GetGeoid	Indicates the Geoid computations should be performed.
ELEVATION_PROCESS=E_CalcTrop	Indicates the tropospheric computations should be performed.
ELEVATION_PROCESS=E_IntrpPOD	Indicates the POD computations should be performed.
ELEVATION_PROCESS=E_CalcStdIR	Indicates the standard range computations should be performed.
ELEVATION_PROCESS=E_CalcLdIR	Indicates the land-specific range computations should be performed.
ELEVATION_PROCESS=E_CalcOcIRc	Indicates the ocean-specific range computations should be performed.
ELEVATION_PROCESS=E_CalcSiIR	Indicates the sea ice-specific range computations should be performed.
ELEVATION_PROCESS=E_CalcIsIR	Indicates the ice sheet-specific range computations should be performed.
ELEVATION_PROCESS=E_CalcSpLoc	Indicates the spot location computations should be performed.
ELEVATION_PROCESS=E_AtmQF	
ELEVATION_PROCESS=E_CalcSlope	Indicates the slope computations should be performed.
ELEVATION_PROCESS=E_CalcRefl	Indicates the reflectance computations should be performed.
ELEVATION_PROCESS=E_ChckReg	Indicates the region-specific checks should be performed.
ELEVATION_PROCESS=E_CalcRegRng	Indicates the regional range computations should be performed.
ELEVATION_PROCESS=E_CalcRegParm	Indicates the region parameter computations should be performed.
ELEVATION_PROCESS=E_CalcDEM	Indicates the DEM computations should be performed.
ELEVATION_PROCESS=ALL	Indicates that GLAS_Alt should perform all elevation processing.

Combinations of the execution flags are allowed. For example, to perform both waveform and elevation processing, the control file would have the following entries:

```
WAVEFORM_PROCESS=ALL
ELEVATION_PROCESS=ALL
```

A control file option is available to designate that location information be taken from the input data rather than the ANC04/08/09 geolocation file. This is useful for quick-look processing when the geolocation files are not yet available but should not be used for normal, production processing. In addition, this option really does not make sense for elevation processing since elevation is highly location and range dependent.

To indicate that pass-thru processing should be used, add the following line to the control file:

```
POSITION=PASS_THRU
```

If this line is not included in the control file, geolocation processing will occur by default.

8.3 Output Files

Table 8-3 lists the outputs created by GLAS_Alt. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the output files

Granule boundaries are designated by specifying start and stop times on the OUTPUT_FILE control file entries.

Table 8-3 GLAS_Alt Outputs

File Spec	Type	Destination	Short Description
gla05*.dat	L1B Alt Product	Archive/GLAS_Alt	The level 1B waveform parameterization product file. Contains the output from the waveform characterization procedure and other parameters required to calculate surface slope and relief characteristics.
gla06*.dat	L2 Alt Product	Archive/GLAS_Alt	L1B elevation data product file. Contains the surface elevation, surface roughness assuming no slope, surface slope assuming no roughness and geodetic and atmospheric corrections for the range.

Table 8-3 GLAS_Alt Outputs (Continued)

File Spec	Type	Destination	Short Description
gla12*.dat	L2 Alt Product	Archive	L2 ice sheet altimetry product file. Contains the ice sheet elevation and elevation distribution calculated from algorithms fine-tuned for ice sheet returns.
gla13*.dat	L2 Alt Product	Archive	L2 sea ice altimetry product file. Contains the sea ice freeboard and sea ice roughness calculated from algorithms fine-tuned for sea ice returns.
gla14*.dat	L2 Alt Product	Archive	L2 land altimetry product file. Contains the land elevation and land elevation distribution calculated from algorithms fine-tuned for land returns.
gla15*.dat	L2 Alt Product	Archive	L2 ocean altimetry product file. Contains ocean elevation and small-scale roughness calculated from algorithms fine-tuned for ocean returns.
gla05*.qap	L1B Alt Quality	QA	L1B waveform parameterization quality file.
gla06*.qap	L2 Alt Quality	QA	L2 elevation data quality file.
gla12*.qap	L2 Alt Quality	QA	L2 ice sheet altimetry quality file.
gla13*.qap	L2 Alt Quality	QA	L2 sea ice altimetry quality file.
gla14*.qap	L2 Alt Quality	QA	L2 land altimetry quality file.
gla15*.qap	L2 Alt Quality	QA	L2 ocean altimeter quality file.
anc06*.txt	Dynamic Ancillary	ISIPS Operations	Standard metadata/processing log file.

Of importance is the relationship between the execution flags and the output products. If a particular execution flag is present in the control file, then the corresponding output file type must be present as well. Table 8-4 shows the relationship between output products and execution flags These relationships will be more fully defined in the next section.

8.4 Processing and Reprocessing Scenarios

GLAS_Alt supports full and partial processing and reprocessing scenarios. Control file entries are show for each supported scenario and define the requisite execution

Table 8-4 GLAS_Alt Execution Flags and Output Products

Execution Flag	Required Output Product
WAVEFORM_PROCESS=ALL	gla05*.dat, gla05*.qap
ELEVATION_PROCESS=ALL	gla06*.dat, gla06*.qap, gla12*.dat, gla12*.qap, gla13*.dat, gla13*.qap, gla14*.dat, gla14*.qa[, gla15*.dat, gla15*.qap

flags as well as input and output files. Only those control file entries unique to the particular processing scenario are shown for brevity. Additionally, even though multiple files of the same time may be needed, a single instance is shown.

Time-selected processing may also be performed by specifying specific start and stop times on the INPUT_FILE and OUTPUT_FILE control file entries.

8.4.1 Full Processing

This scenario is used to create both waveform and elevation products from GLA01 and the requisite ANC files. All files listed in Table 8-1 and Table 8-3 are required.

8.4.2 Partial Processing: Waveform Only

This scenario is used to process only Waveform data, creating GLA05 data from selected GLA01 products and the requisite ANC files. All files listed in Table 8-1 and Table 8-3, except those listed as Elevation-only, are required. The following execution flag is required:

```
WAVEFORM_PROCESS=ALL
```

8.4.3 Partial Processing/Re-Processing: Elevation Only

This scenario is used to process only Elevation data, creating GLA06 and GLA12-15 data from selected GLA05 products (created during a previous execution of GLAS_Alt) and the requisite ANC files. All files listed in Table 8-1 and Table 8-3, except those listed as Waveform-only, are required. The following execution flag is required:

```
ELEVATION_PROCESS=ALL
```

8.4.4 Predict Orbit Processing: Waveform Only

This scenario is used to quickly process Waveform data when only predicted orbit (ANC20) is available. The input files listed in Table 8-5 are required.

Table 8-5 GLAS_Alt Predict Waveform Inputs

File Spec	Type	Source	Short Description
anc07*_0000.dat	Static Ancillary	Science Team	Error file.
anc07*_0001.dat	Static Ancillary	Science Team	Global constants file.
anc07*_0003.dat	Static Ancillary	Science Team	Waveform constants file.
anc20*.dat	Dynamic Ancillary	UTexas	Predict Orbit file.

Table 8-5 GLAS_Alt Predict Waveform Inputs (Continued)

File Spec	Type	Source	Short Description
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc27*_0000.dat	Static Ancillary	Science Team	Coarse regional mask file.
anc27*_0001.dat	Static Ancillary	Science Team	Fine regional mask file.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.
anc45*_0001.dat	Static Ancillary	Science Team	GLA01 metadata input file. *Waveform only
anc45*_0005.dat	Static Ancillary	Science Team	GLA05 metadata input file.
Control File	Control	ISIPS Operations	Control file.
gla01*_.dat	Level-1A Product	GLAS_L1A	L1A Altimetry product file.

8.4.5 Predict Orbit Processing: Elevation Only

This scenario is used to quickly process Elevation data when only predicted orbit (ANC20) is available. The input files listed in Table 8-6 are required.

Table 8-6 GLAS_Alt Predict Elevation Inputs

File Spec	Type	Source	Short Description
anc07*_0000.dat	Static Ancillary	Science Team	Error file.
anc07*_0001.dat	Static Ancillary	Science Team	Global constants file.
anc07*_0004.dat	Static Ancillary	Science Team	Elevations constant file.
anc12*_0000.dat	Static Ancillary	Science Team	Coarse DEM file.
anc12*_0001.dat	Static Ancillary	Science Team	Fine DEM file
anc13*.dat	Static Ancillary	Science Team	Geoid file
anc16*.dat	Static Ancillary	Science Team	Load tide coefficients file
anc17*.dat	Static Ancillary	Science Team	Ocean tide coefficients file
anc20*.dat	Dynamic Ancillary	UTexas	Predict Orbit file.
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc27*_0000.dat	Static Ancillary	Science Team	Coarse regional mask file.
anc27*_0001.dat	Static Ancillary	Science Team	Fine regional mask file.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.
anc41*.dat	Dynamic Ancillary	Science Team	JPL Planetary Ephemeris
anc45*_0005.dat	Static Ancillary	Science Team	GLA05 metadata input file.
anc45*_0006.dat	Static Ancillary	Science Team	GLA06 metadata input file.

Table 8-6 GLAS_Alt Predict Elevation Inputs (Continued)

File Spec	Type	Source	Short Description
anc45*_0012.dat	Static Ancillary	Science Team	GLA12 metadata input file.
anc45*_0013.dat	Static Ancillary	Science Team	GLA13 metadata input file.
anc45*_0014.dat	Static Ancillary	Science Team	GLA14 metadata input file.
anc45*_0015.dat	Static Ancillary	Science Team	GLA15 metadata input file.
Control File	Control	ISIPS Operations	Control file.
gla05*_.dat	Level-1B Product	GLAS_Alt	L1B Waveform product file.
gla09*_.dat	Level-2 Product	GLAS_Atm	L2 Atmosphere product file. (* Optional)
gla11*_.dat	Level-2 Product	GLAS_Atm	L2 Atmosphere product file. (* Optional)

The following control flags are required:

```
WAVEFORM_PROCESS=NONE
ELEVATION_PROCESS=E_IntrpPOD
ELEVATION_PROCESS=E_CalcDEM
ELEVATION_PROCESS=E_CalcSlope
ELEVATION_PROCESS=E_CalcLoadTD
ELEVATION_PROCESS=E_CalcOceanTD
ELEVATION_PROCESS=E_CalcEarthTD
ELEVATION_PROCESS=E_GetGeoid
SURFACE_TYPE=ALL
```

8.4.6 Re-Processing: Elevation Region-Specific Only

This scenario is used to recreate the GLA12-15 region-specific products from a previously created GLA05/GLA06 and the requisite ANC files. All files listed in Table 8-1 and Table 8-3, except those listed as Waveform-only, are required, with the following exceptions:

- GLA06 is required as an input but not an output.
- Tide files (ANC16, ANC17) are not required.
- The geoid file (ANC13) is not required.
- The meteorological files (ANC01) are not required.

The following execution flags are required:

```
ELEVATION_PROCESS=E_GetGeoid
ELEVATION_PROCESS=E_IntrpPOD
ELEVATION_PROCESS=E_CalcStdIR
ELEVATION_PROCESS=E_CalcLdIR
ELEVATION_PROCESS=E_CalcOcIR
ELEVATION_PROCESS=E_CalcSiIR
ELEVATION_PROCESS=E_CalcIsIR
ELEVATION_PROCESS=E_CALCSpLoc
ELEVATION_PROCESS=E_AtmQF
```

```
ELEVATION_PROCESS=E_CaclSlope
ELEVATION_PROCESS=E_CalcRefl
ELEVATION_PROCESS=E_ChckReg
ELEVATION_PROCESS=E_CalcRegRng
ELEVATION_PROCESS=E_CalcRegParm
ELEVATION_PROCESS=E_CalcDEM
```

8.5 Startup and Termination

This is an overview of the steps necessary to run GLAS_Alt.

8.5.1 Setup a Runtime Directory

The suggested method of running GLAS_Alt is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a GLAS_Alt run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_Alt .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc*.dat .
ln -s $DATA_DIR/*.dat .

8.5.2 Create a Control File

GLAS_Alt is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

cp $GLAS_HOME/data/cf_glas_alt_(*wf*|*elev*).ctl ./*control_file_name*
vi ./*control_file_name*

8.5.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

./GLAS_Alt *control_file_name*

8.5.4 Run-Time Statistics

GLAS_Alt processes 25 hours of GLAS data in approximately 12 hours for full waveform processing. GLAS_Alt processes 25 hours of GLAS data in approximately 1.5 hours for full elevation processing. This may vary due to hardware load and data dependencies.

8.5.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

8.6 Error and Warning Messages

All GSAS error and warning messages are numerically listed in Appendix D.

8.7 Recovery Steps

If GLAS_Alt terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem. Refer to errors listed in Appendix D for assistance in determining the problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_Alt will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions.

8.8 Sample GLAS_Alt Waveform Control File

```
=GLAS_ALT
PGE_VERSION=2.2
PASSID=1 22010052332 51860152 51865951 123.9248348 4771
PASSID=2 22010052333 51865951 51871750  99.7298577 4772
PASSID=3 22010052334 51871750 51877549  75.5345768 4773
PASSID=4 22010052335 51877549 51883348  51.3401797 4774
INPUT_FILE=anc07_001_01_0000.dat 0 0
INPUT_FILE=anc07_001_01_0001.dat 0 0
INPUT_FILE=anc07_001_01_0002.dat 0 0
INPUT_FILE=anc07_001_01_0003.dat 0 0
INPUT_FILE=anc07_001_01_0004.dat 0 0
INPUT_FILE=anc07_001_01_0006.dat 0 0
INPUT_FILE=anc07_001_01_0005.dat 0 0
INPUT_FILE=ANC08_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC08_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=ANC04_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC04_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=ANC09_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC09_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=anc45_001_01_0001.dat 0 0
INPUT_FILE=anc45_001_01_0002.dat 0 0
INPUT_FILE=anc45_001_01_0003.dat 0 0
INPUT_FILE=anc45_001_01_0004.dat 0 0
INPUT_FILE=anc45_001_01_0005.dat 0 0
INPUT_FILE=anc45_001_01_0006.dat 0 0
INPUT_FILE=anc45_001_01_0007.dat 0 0
INPUT_FILE=anc45_001_01_0008.dat 0 0
```

```
INPUT_FILE=anc45_001_01_0009.dat 0 0
INPUT_FILE=anc45_001_01_0010.dat 0 0
INPUT_FILE=anc45_001_01_0011.dat 0 0
INPUT_FILE=anc45_001_01_0012.dat 0 0
INPUT_FILE=anc45_001_01_0013.dat 0 0
INPUT_FILE=anc45_001_01_0014.dat 0 0
INPUT_FILE=anc45_001_01_0015.dat 0 0
INPUT_FILE=anc27_001_01_0000.dat 0 0
INPUT_FILE=anc27_001_01_0001.dat 0 0
#
INPUT_FILE=anc25_00_20020406_000000_00_0000.dat 51904800 51926400
INPUT_FILE=anc33_00_20020406_000000_00_0000.dat 51904800 51926400
#
INPUT_FILE=GLA01_002_22_01005_2332_1_01_0000.DAT 51860959 51862240 2 1 1
INPUT_FILE=GLA01_002_22_01005_2332_2_01_0000.DAT 51862240 51863856 2 1 2
INPUT_FILE=GLA01_002_22_01005_2332_3_01_0000.DAT 51863856 51865142 2 1 3
INPUT_FILE=GLA01_002_22_01005_2332_4_01_0000.DAT 51865142 51866758 2 1 4
INPUT_FILE=GLA01_002_22_01005_2333_1_01_0000.DAT 51866758 51868039 2 1 1
INPUT_FILE=GLA01_002_22_01005_2333_2_01_0000.DAT 51868039 51869655 2 1 2
INPUT_FILE=GLA01_002_22_01005_2333_3_01_0000.DAT 51869655 51870941 2 1 3
INPUT_FILE=GLA01_002_22_01005_2333_4_01_0000.DAT 51870941 51872557 2 1 4
INPUT_FILE=GLA01_002_22_01005_2334_1_01_0000.DAT 51872557 51873837 2 1 1
INPUT_FILE=GLA01_002_22_01005_2334_2_01_0000.DAT 51873837 51875453 2 1 2
INPUT_FILE=GLA01_002_22_01005_2334_3_01_0000.DAT 51875453 51876740 2 1 3
INPUT_FILE=GLA01_002_22_01005_2334_4_01_0000.DAT 51876740 51878356 2 1 4
#
OUTPUT_FILE=GLA05_002_22_01005_2332_1_01_0000.DAT 51860959 51862240 2 1 1
OUTPUT_FILE=GLA05_002_22_01005_2332_2_01_0000.DAT 51862240 51863856 2 1 2
OUTPUT_FILE=GLA05_002_22_01005_2332_3_01_0000.DAT 51863856 51865142 2 1 3
OUTPUT_FILE=GLA05_002_22_01005_2332_4_01_0000.DAT 51865142 51866758 2 1 4
OUTPUT_FILE=GLA05_002_22_01005_2333_1_01_0000.DAT 51866758 51868039 2 1 1
OUTPUT_FILE=GLA05_002_22_01005_2333_2_01_0000.DAT 51868039 51869655 2 1 2
OUTPUT_FILE=GLA05_002_22_01005_2333_3_01_0000.DAT 51869655 51870941 2 1 3
OUTPUT_FILE=GLA05_002_22_01005_2333_4_01_0000.DAT 51870941 51872557 2 1 4
OUTPUT_FILE=GLA05_002_22_01005_2334_1_01_0000.DAT 51872557 51873837 2 1 1
OUTPUT_FILE=GLA05_002_22_01005_2334_2_01_0000.DAT 51873837 51875453 2 1 2
OUTPUT_FILE=GLA05_002_22_01005_2334_3_01_0000.DAT 51875453 51876740 2 1 3
OUTPUT_FILE=GLA05_002_22_01005_2334_4_01_0000.DAT 51876740 51878356 2 1 4
OUTPUT_FILE=QAP05_002_22_01005_2332_1_01_0000.DAT 51860959 51862240 2 1 1
OUTPUT_FILE=QAP05_002_22_01005_2332_2_01_0000.DAT 51862240 51863856 2 1 2
OUTPUT_FILE=QAP05_002_22_01005_2332_3_01_0000.DAT 51863856 51865142 2 1 3
OUTPUT_FILE=QAP05_002_22_01005_2332_4_01_0000.DAT 51865142 51866758 2 1 4
OUTPUT_FILE=QAP05_002_22_01005_2333_1_01_0000.DAT 51866758 51868039 2 1 1
OUTPUT_FILE=QAP05_002_22_01005_2333_2_01_0000.DAT 51868039 51869655 2 1 2
OUTPUT_FILE=QAP05_002_22_01005_2333_3_01_0000.DAT 51869655 51870941 2 1 3
OUTPUT_FILE=QAP05_002_22_01005_2333_4_01_0000.DAT 51870941 51872557 2 1 4
OUTPUT_FILE=QAP05_002_22_01005_2334_1_01_0000.DAT 51872557 51873837 2 1 1
OUTPUT_FILE=QAP05_002_22_01005_2334_2_01_0000.DAT 51873837 51875453 2 1 2
OUTPUT_FILE=QAP05_002_22_01005_2334_3_01_0000.DAT 51875453 51876740 2 1 3
OUTPUT_FILE=QAP05_002_22_01005_2334_4_01_0000.DAT 51876740 51878356 2 1 4
OUTPUT_FILE=ANC06_001_20010823_180000_01_004533_GLAS_ALT.TXT 51861600 51883201
WAVEFORM_PROCESS=ALL
ELEVATION_PROCESS=NONE
```

8.9 Sample GLAS_Alt Elevation Control File

```
=GLAS_ALT
PGE_VERSION=2.2
```

```
PASSID=1 22010052332 51860152 51865951 123.9248348 4771
PASSID=2 22010052333 51865951 51871750  99.7298577 4772
PASSID=3 22010052334 51871750 51877549  75.5345768 4773
PASSID=4 22010052335 51877549 51883348  51.3401797 4774
INPUT_FILE=anc07_001_01_0000.dat 0 0
INPUT_FILE=anc07_001_01_0001.dat 0 0
INPUT_FILE=anc07_001_01_0002.dat 0 0
INPUT_FILE=anc07_001_01_0003.dat 0 0
INPUT_FILE=anc07_001_01_0004.dat 0 0
INPUT_FILE=anc07_001_01_0006.dat 0 0
INPUT_FILE=anc07_001_01_0005.dat 0 0
INPUT_FILE=ANC08_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC08_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=ANC04_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC04_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=ANC09_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC09_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=anc45_001_01_0001.dat 0 0
INPUT_FILE=anc45_001_01_0002.dat 0 0
INPUT_FILE=anc45_001_01_0003.dat 0 0
INPUT_FILE=anc45_001_01_0004.dat 0 0
INPUT_FILE=anc45_001_01_0005.dat 0 0
INPUT_FILE=anc45_001_01_0006.dat 0 0
INPUT_FILE=anc45_001_01_0007.dat 0 0
INPUT_FILE=anc45_001_01_0008.dat 0 0
INPUT_FILE=anc45_001_01_0009.dat 0 0
INPUT_FILE=anc45_001_01_0010.dat 0 0
INPUT_FILE=anc45_001_01_0011.dat 0 0
INPUT_FILE=anc45_001_01_0012.dat 0 0
INPUT_FILE=anc45_001_01_0013.dat 0 0
INPUT_FILE=anc45_001_01_0014.dat 0 0
INPUT_FILE=anc45_001_01_0015.dat 0 0
INPUT_FILE=anc27_001_01_0000.dat 0 0
INPUT_FILE=anc27_001_01_0001.dat 0 0
# DEM Files
INPUT_FILE=anc12_000_00_0000.dat  0 51926400
INPUT_FILE=anc12_000_00_0001.dat  0 51926400
# Geoid File
INPUT_FILE=anc13_001_01_0000.dat 00 51926400
# Load Tide File
INPUT_FILE=anc16_001_01_0000.dat 51861600 51926400
# Ocean Tide File
INPUT_FILE=anc17_001_01_0000.dat 51861600 51926400
INPUT_FILE=ANC01_000_20010823_120000_01_0000.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_120000_01_0001.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_120000_01_0002.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_120000_01_0003.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_120000_01_0004.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_180000_01_0000.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010823_180000_01_0001.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010823_180000_01_0002.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010823_180000_01_0003.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010823_180000_01_0004.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010824_000000_01_0000.DAT  51904800 51926400
INPUT_FILE=ANC01_000_20010824_000000_01_0001.DAT  51904800 51926400
INPUT_FILE=ANC01_000_20010824_000000_01_0002.DAT  51904800 51926400
INPUT_FILE=ANC01_000_20010824_000000_01_0003.DAT  51904800 51926400
INPUT_FILE=ANC01_000_20010824_000000_01_0004.DAT  51904800 51926400
#
```

```
INPUT_FILE=anc25_00_20020406_000000_00_0000.dat 51904800 51926400
INPUT_FILE=anc33_00_20020406_000000_00_0000.dat 51904800 51926400
#
INPUT_FILE=GLA05_002_22_01005_2332_1_01_0000.DAT 51860959 51862240 2 1 1
INPUT_FILE=GLA05_002_22_01005_2332_2_01_0000.DAT 51862240 51863856 2 1 2
INPUT_FILE=GLA05_002_22_01005_2332_3_01_0000.DAT 51863856 51865142 2 1 3
INPUT_FILE=GLA05_002_22_01005_2332_4_01_0000.DAT 51865142 51866758 2 1 4
INPUT_FILE=GLA05_002_22_01005_2333_1_01_0000.DAT 51866758 51868039 2 1 1
INPUT_FILE=GLA05_002_22_01005_2333_2_01_0000.DAT 51868039 51869655 2 1 2
INPUT_FILE=GLA05_002_22_01005_2333_3_01_0000.DAT 51869655 51870941 2 1 3
INPUT_FILE=GLA05_002_22_01005_2333_4_01_0000.DAT 51870941 51872557 2 1 4
INPUT_FILE=GLA05_002_22_01005_2334_1_01_0000.DAT 51872557 51873837 2 1 1
INPUT_FILE=GLA05_002_22_01005_2334_2_01_0000.DAT 51873837 51875453 2 1 2
INPUT_FILE=GLA05_002_22_01005_2334_3_01_0000.DAT 51875453 51876740 2 1 3
INPUT_FILE=GLA05_002_22_01005_2334_4_01_0000.DAT 51876740 51878356 2 1 4
#
OUTPUT_FILE=GLA06_002_22_01005_2332_1_00_0000.DAT 51860959 51862240 2 1 1
OUTPUT_FILE=GLA06_002_22_01005_2332_2_01_0000.DAT 51862240 51863856 2 1 2
OUTPUT_FILE=GLA06_002_22_01005_2332_3_01_0000.DAT 51863856 51865142 2 1 3
OUTPUT_FILE=GLA06_002_22_01005_2332_4_01_0000.DAT 51865142 51866758 2 1 4
OUTPUT_FILE=GLA06_002_22_01005_2333_1_01_0000.DAT 51866758 51868039 2 1 1
OUTPUT_FILE=GLA06_002_22_01005_2333_2_01_0000.DAT 51868039 51869655 2 1 2
OUTPUT_FILE=GLA06_002_22_01005_2333_3_01_0000.DAT 51869655 51870941 2 1 3
OUTPUT_FILE=GLA06_002_22_01005_2333_4_01_0000.DAT 51870941 51872557 2 1 4
OUTPUT_FILE=GLA06_002_22_01005_2334_1_01_0000.DAT 51872557 51873837 2 1 1
OUTPUT_FILE=GLA06_002_22_01005_2334_2_01_0000.DAT 51873837 51875453 2 1 2
OUTPUT_FILE=GLA06_002_22_01005_2334_3_01_0000.DAT 51875453 51876740 2 1 3
OUTPUT_FILE=GLA06_002_22_01005_2334_4_01_0000.DAT 51876740 51878356 2 1 4
OUTPUT_FILE=GLA06_002_22_01005_2332_1_00_0000.QAP 51860959 51862240 2 1 1
OUTPUT_FILE=GLA06_002_22_01005_2332_2_01_0000.QAP 51862240 51863856 2 1 2
OUTPUT_FILE=GLA06_002_22_01005_2332_3_01_0000.QAP 51863856 51865142 2 1 3
OUTPUT_FILE=GLA06_002_22_01005_2332_4_01_0000.QAP 51865142 51866758 2 1 4
OUTPUT_FILE=GLA06_002_22_01005_2333_1_01_0000.QAP 51866758 51868039 2 1 1
OUTPUT_FILE=GLA06_002_22_01005_2333_2_01_0000.QAP 51868039 51869655 2 1 2
OUTPUT_FILE=GLA06_002_22_01005_2333_3_01_0000.QAP 51869655 51870941 2 1 3
OUTPUT_FILE=GLA06_002_22_01005_2333_4_01_0000.QAP 51870941 51872557 2 1 4
OUTPUT_FILE=GLA06_002_22_01005_2334_1_01_0000.QAP 51872557 51873837 2 1 1
OUTPUT_FILE=GLA06_002_22_01005_2334_2_01_0000.QAP 51873837 51875453 2 1 2
OUTPUT_FILE=GLA06_002_22_01005_2334_3_01_0000.QAP 51875453 51876740 2 1 3
OUTPUT_FILE=GLA06_002_22_01005_2334_4_01_0000.QAP 51876740 51878356 2 1 4
OUTPUT_FILE=GLA12_002_22_01005_2325_0_01_0000.DAT 51820367 51878356 2 1 0
OUTPUT_FILE=GLA13_002_22_01005_2325_0_01_0000.DAT 51820367 51878356 2 1 0
OUTPUT_FILE=GLA14_002_22_01005_2325_0_01_0000.DAT 51820367 51878356 2 1 0
OUTPUT_FILE=GLA15_002_22_01005_2325_0_01_0000.DAT 51820367 51878356 2 1 0
OUTPUT_FILE=GLA12_002_22_01005_2325_0_01_0000.QAP 51820367 51878356 2 1 0
OUTPUT_FILE=GLA13_002_22_01005_2325_0_01_0000.QAP 51820367 51878356 2 1 0
OUTPUT_FILE=GLA14_002_22_01005_2325_0_01_0000.QAP 51820367 51878356 2 1 0
OUTPUT_FILE=GLA15_002_22_01005_2325_0_01_0000.QAP 51820367 51878356 2 1 0
OUTPUT_FILE=ANC06_001_20010823_180000_01_0032420_GLAS_ALT.TXT 51861600 51883201
ELEVATION_PROCESS=ALL
```

Section 9
GLAS_Atm

9.1 Function

GLAS_Atm uses algorithms defined in the GLAS Atmosphere ATBD to first produce calibrated backscatter cross section and then to locate all radiatively significant cloud and aerosol features present in the data and define their height and optical depth. This is done by first locating the top and bottom of the layers, differentiating between cloud and aerosol and then computing the extinction and optical depth of each layer. The layers are classified as either 1) cloud, 2) elevated aerosol, or 3) planetary boundary layer.

In its most basic execution, GLAS_Atm reads GLAS Level-1A GLA02 files and requisite ancillary files to create GLAS L1B and 2 Atmosphere product files (GLA07-11). However, in some execution scenarios, GLAS_Atm can read the GLA07 file created in a previous execution to produce GLA08-11 products.GLAS_Atm uses the ANC07 files for constants and error/status messages. It uses the ANC12 DEM files to determine a preliminary elevation from a global Digital Elevation Model. It uses the ANC13 file to determine the geoid, ANC18 for reference standard atmosphere measurements and ANC01 for meteorological information. It uses ANC09 for precision attitude, and ANC04/ANC08 for precision orbit information and ANC45 files for initializing metadata information. It uses ANC36 files generated by atm_anc to perform calibrations and ANC38 to perform multiple scattering corrections. ANC25 and ANC33 files are used for time delay and oscillator-based corrections.

GLAS_Atm should be run once after successful execution of GLAS_L1A, atm_anc, and met_util. GLAS_Atm permits multi-granule input/output files, input/output time selection, selected execution, and reprocessing.

9.2 Input Files

Table 9-1 lists the inputs files to GLAS_Atm. Files which are specific to GLAS_Atm are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the non-specific files.

9.2.1 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_Atm are described here.

The control file section delimiter for GLAS_Atm is:

```
=GLAS_Atm
```

In order to satisfy the partial-processing/reprocessing requirement, GLAS_Atm can perform limited processing based on execution flags within the control file. The flags and the processes they control are defined in Table 9-2.

Table 9-1 GLAS_Atm Inputs

File Spec	Type	Source	Short Description
anc01*.dat	Dynamic Ancillary	met_util	Meteorological subset files. Data sets at times before and after the time of the profile are interpolated to the time of the profile. If either of the ANC01 data sets are missing, then the available ANC01 data set is used without interpolation. If no ANC01 data sets are available, then standard atmosphere data are used instead.
anc04*.dat	Dynamic Ancillary	UTexas	IERS Polar Motion and Earth Rotation Data File.
anc07*_0000.dat	Static Ancillary	Science Team	Error file.
anc07*_0001.dat	Static Ancillary	Science Team	Global constants file.
anc07*_0002.dat	Static Ancillary	Science Team	Atm constants file.
anc07*_0005.dat	Static Ancillary	Science Team	L1A constants file.
anc08*.dat	Dynamic Ancillary	UTexas	Precision Orbit file.
anc09*.dat	Dynamic Ancillary	UTexas	Precision Attitude file.
anc12*_0000.dat	Static Ancillary	Science Team	DEM file.
anc12*_0001.dat	Static Ancillary	Science Team	DEM mask file.
anc13*.dat	Static Ancillary	Science Team	Geoid file.
anc18*.dat	Static Ancillary	Science Team	Standard atmosphere file.
anc24*.dat	Dynamic Ancillary	UTexas	Rotation Matrix file.
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc30*.dat	Static Ancillary	Science Team	Global aerosol categorization map file.
anc31*.dat	Static Ancillary	Science Team	Aerosol tropospheric classification map file.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.
anc35*.dat	Static Ancillary	Science Team	Ozone file.
anc36*.dat	Dynamic Ancillary	atm_anc	Atmosphere Calibration file.
anc38*.dat	Static Ancillary	Science Team	Multiple-scattering table file.
anc41*.dat	Dynamic Ancillary	Science Team	JPL Planetary Ephemeris
anc45*_0002.dat	Static Ancillary	Science Team	GLA02 metadata input file.

Table 9-1 GLAS_Atm Inputs

File Spec	Type	Source	Short Description
anc45*_0007.dat	Static Ancillary	Science Team	GLA07 metadata input file.
anc45*_0008.dat	Static Ancillary	Science Team	GLA08 metadata input file.
anc45*_0009.dat	Static Ancillary	Science Team	GLA09 metadata input file.
anc45*_0010.dat	Static Ancillary	Science Team	GLA10 metadata input file.
anc45*_0011.dat	Static Ancillary	Science Team	GLA11 metadata input file.
Control File	Control	ISIPS Operations	Control file.
gla02*_.dat	Level-1A Product	GLAS_L1A	L1A Atmosphere product file.

Table 9-2 GLAS_Atm Control Flags

Flag	Significance
ATMOSPHERE_PROCESS=None	Indicates that GLAS_Atm should perform no processing.
ATMOSPHERE_PROCESS=A_bs_to_end	Indicates that GLAS_Atm should perform all processing from backscatter-to-end.
ATMOSPHERE_PROCESS=A_bs_only	Indicates that GLAS_Atm should perform backscatter-only processing.
ATMOSPHERE_PROCESS=A_cld_to_end	Indicates that GLAS_Atm should perform processing from cloud-to-end.
ATMOSPHERE_PROCESS=A_no_pod	Indicates that GLAS_Atm should not perform precision orbit determination.
ATMOSPHERE_PROCESS=ALL	Indicates that GLAS_Atm should perform all processing from backscatter-to-end, including precision orbit determination.

Combinations of the execution flags are allowed. For example, to perform from backscatter to end and no POD, the control file would have the following entries:

```
ATMOSPHERE_PROCESS=A_bs_to_end
ATMOSPHERE_PROCESS=A_no_pod
```

A control file option is available to designate that location information be taken from the input data rather than the ANC04/08/09 geolocation file. This is useful for quick-look processing when the geolocation files are not yet available.

To indicate that pass-thru processing should be used, add the following line to the control file:

```
POSITION=PASS_THRU
```

If this line is not included in the control file, geolocation processing will occur by default.

9.3 Output Files

Table 9-3 lists the outputs created by GLAS_Atm. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the output files

Granule boundaries are designated by specifying start and stop times on the OUTPUT_FILE control file entries.

Table 9-3 GLAS_Atm Outputs

File Spec	Type	Destination	Short Description
gla07*.dat	L1B Atm Product	Archive	L1B Global Backscatter product file. Contains full 532 nm and 1064 nm calibrated attenuated backscatter profiles at 5 times per second, and from 10 to -1 km, at 40 times per second. Also included will be calibration coefficient values and molecular backscatter profiles at once per second.
gla08*.dat	L2 Atm Product	Archive	L2 Planetary Boundary Layer and Elevated Aerosol Layer Height product file. Contains elevated aerosol layer height data consisting of top and bottom heights for up to 5 aerosol layers below 20 km at once per 4 seconds, and top and bottom heights for up to 3 aerosol layers above 20 km at once per 20 seconds.The boundary layer height is provided at once per 4 second and 5 times per second resolution.
gla09*.dat	L2 Atm Product	Archive	L2 Cloud Layer Height product file. Contains top and bottom heights for up to 10 layers below 20 km at once per 4 seconds, once per second, 5 times per second, and 40 times per second (below 4 km only). Ground heights will also be provided at each resolution.

Table 9-3 GLAS_Atm Outputs

File Spec	Type	Destination	Short Description
gla10*.dat	L2 Atm Product	Archive	L2 Aerosol Vertical Structure product file. Contains cloud and aerosol backscatter and extinction cross section profiles.
gla11*.dat	L2 AtmProduct	Archive	L2 Thin Cloud/Aerosol product file. Contains optical depths for clouds for up to 10 layers, the planetary boundary layer, and aerosols for up to 8 layers.Cloud optical depth is at once per second and aerosol optical depth is at once per 4 second resolution.
gla07*.qap	L2 Atm Quality	QA	L1B Global Backscatter quality file.
gla08*.qap	L2 Atm Quality	QA	L2 Planetary Boundary Layer and Elevated Aerosol Layer Height quality file.
gla09*.qap	L2 Atm Quality	QA	L2 Cloud Layer Height quality file.
gla10*.qap	L2 Atm Quality	QA	L2 Aerosol Vertical Structure quality file.
gla11*.qap	L2 Atm Quality	QA	L2 Thin Cloud/Aerosol quality file.
anc06*.txt	Dynamic Ancillary	ISIPS Operations	Standard metadata/processing log file.

Of importance is the relationship between the execution flags and the output products. If a particular execution flag is present in the control file, then the corresponding output file type must be present as well. Table 9-4 shows the relationship between output products and execution flags.These relationships will be more fully defined in the next section.

Table 9-4 GLAS_Atm Execution Flags and Output Products

Execution Flag	Required Output Product
ATMOSPHERE_PROCESS=A_bs_to_end	gla07*.dat, qap07*.dat, gla08*.dat, qap08*.dat, gla09*.dat, qap09*dat, gla10*.dat, qap10*.dat, gla11*.dat qap11*.dat
ATMOSPHERE_PROCESS=A_bs_only	gla07*.dat, qap07*.dat,
ATMOSPHERE_PROCESS=A_cld_to_end	gla08*.dat, qap08*.dat, gla09*.dat, qap09*dat, gla10*.dat, qap10*.dat, gla11*.dat qap11*.dat

9.4 Processing and Reprocessing Scenarios

GLAS_Atm supports full and partial processing scenarios. Control file entries are show for each supported scenario and define the requisite execution flags as well as input and output files. Only those control file entries unique to the particular processing scenario are show for brevity. Additionally, even though multiple files of the same time may be needed, a single instance is show.

Time-selected processing may also be performed by specifying specific start and stop times on the INPUT_FILE and OUTPUT_FILE control file entries.

9.4.1 Standard Processing

This scenario is used for normal Atmosphere processing, creating GLA07, GLA08, GLA09, GLA10 and GLA11 data from GLA02 and the requisite ANC files. All files listed in Table 9-1 and Table 9-3 are required. The following execution flag is required:

```
ATMOSPHERE_PROCESS=A_bs_to_end
```

9.4.2 Predict Orbit Processing

This scenario is used to quickly process Atmosphere data when only predicted orbit (ANC20) is available. The input files listed in Table 9-5 are required.

Table 9-5 GLAS_Atm Predict Inputs

File Spec	Type	Source	Short Description
anc07*_0000.dat	Static Ancillary	Science Team	Error file.
anc07*_0001.dat	Static Ancillary	Science Team	Global constants file.
anc07*_0002.dat	Static Ancillary	Science Team	Atm constants file.
anc07*_0005.dat	Static Ancillary	Science Team	L1A constants file.
anc12*_0000.dat	Static Ancillary	Science Team	DEM file.
anc12*_0001.dat	Static Ancillary	Science Team	DEM mask file.
anc13*.dat	Static Ancillary	Science Team	Geoid file.
anc18*.dat	Static Ancillary	Science Team	Standard atmosphere file.
anc20*.dat	Dynamic Ancillary	UTexas	Predict Orbit file.
anc24*.dat	Dynamic Ancillary	UTexas	Rotation Matrix file.
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc30*.dat	Static Ancillary	Science Team	Global aerosol categorization map file.
anc31*.dat	Static Ancillary	Science Team	Aerosol tropospheric classification map file.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.

Table 9-5 GLAS_Atm Predict Inputs

File Spec	Type	Source	Short Description
anc35*.dat	Static Ancillary	Science Team	Ozone file.
anc38*.dat	Static Ancillary	Science Team	Multiple-scattering table file.
anc41*.dat	Dynamic Ancillary	Science Team	JPL Planetary Ephemeris
anc45*_0002.dat	Static Ancillary	Science Team	GLA02 metadata input file.
anc45*_0007.dat	Static Ancillary	Science Team	GLA07 metadata input file.
anc45*_0008.dat	Static Ancillary	Science Team	GLA08 metadata input file.
anc45*_0009.dat	Static Ancillary	Science Team	GLA09 metadata input file.
anc45*_0010.dat	Static Ancillary	Science Team	GLA10 metadata input file.
anc45*_0011.dat	Static Ancillary	Science Team	GLA11 metadata input file.
Control File	Control	ISIPS Operations	Control file.
gla02*_.dat	Level-1A Product	GLAS_L1A	L1A Atmosphere product file.

9.4.3 Partial Processing: Backscatter

This scenario is used for backscatter-only Atmosphere processing, creating GLA07 data from GLA02 and the requisite ANC files. All files listed in Table 9-1 and Table 9-3 are required, with the following exceptions:

- GLA08-11 output files are not required.
- QAP08-11 output files are not required.

The following execution flag is required:

```
ATMOSPHERE_PROCESS=A_bs_only
```

9.4.4 Partial Processing/Re-Processing: Cloud-to-End

This scenario is used for cloud-to-end Atmosphere-related data, creating GLA08-11 data from selected GLA07 products (created during a previous execution of GLAS_Atm) and the requisite ANC files. All files listed in Table 9-1 and Table 9-3 are required, with the following exception:

- GLA07 is required as an input file, but not an output file.

The following execution flag is required:

```
ATMOSPHERE_PROCESS=A_cld_to_end
```

9.5 Startup and Termination

This is an overview of the steps necessary to run GLAS_Atm.

9.5.1 Setup a Runtime Directory

The suggested method of running GLAS_Atm is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a GLAS_Atm run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_Atm .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc*.dat .
ln -s $DATA_DIR/*.dat .
```

9.5.2 Create a Control File

GLAS_Atm is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

```
cp $GLAS_HOME/data/cf_glas_atm.ctl ./control_file_name
vi ./control_file_name
```

9.5.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

```
./GLAS_Atm control_file_name
```

9.5.4 Run-Time Statistics

GLAS_Atm processes 25 hours of GLAS data in approximately four hours. This may vary due to hardware load and data dependencies.

9.5.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

9.6 Error and Warning Messages

All GSAS error and warning messages are numerically listed in Appendix D.

9.7 Recovery Steps

If GLAS_Atm terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_Atm will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions

9.8 Sample GLAS_Atm Control File

```
=GLAS_ATM
PGE_VERSION=2.2
PASSID=1 22010052332 51860152 51865951 123.9248348 4771
PASSID=2 22010052333 51865951 51871750  99.7298577 4772
PASSID=3 22010052334 51871750 51877549  75.5345768 4773
PASSID=4 22010052335 51877549 51883348  51.3401797 4774
#
INPUT_FILE=anc07_001_01_0000.dat 0 0
INPUT_FILE=anc07_001_01_0001.dat 0 0
INPUT_FILE=anc07_001_01_0002.dat 0 0
INPUT_FILE=anc07_001_01_0003.dat 0 0
INPUT_FILE=anc07_001_01_0004.dat 0 0
INPUT_FILE=anc07_001_01_0006.dat 0 0
INPUT_FILE=anc07_001_01_0005.dat 0 0
INPUT_FILE=ANC08_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC08_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=ANC04_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC04_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=ANC09_001_20010823_000000_02_0000.DAT 51796800 51883200
INPUT_FILE=ANC09_001_20010824_000000_02_0000.DAT 51883200 51969600
INPUT_FILE=anc45_001_01_0001.dat 0 0
INPUT_FILE=anc45_001_01_0002.dat 0 0
INPUT_FILE=anc45_001_01_0003.dat 0 0
INPUT_FILE=anc45_001_01_0004.dat 0 0
INPUT_FILE=anc45_001_01_0005.dat 0 0
INPUT_FILE=anc45_001_01_0006.dat 0 0
INPUT_FILE=anc45_001_01_0007.dat 0 0
INPUT_FILE=anc45_001_01_0008.dat 0 0
INPUT_FILE=anc45_001_01_0009.dat 0 0
INPUT_FILE=anc45_001_01_0010.dat 0 0
INPUT_FILE=anc45_001_01_0011.dat 0 0
INPUT_FILE=anc45_001_01_0012.dat 0 0
INPUT_FILE=anc45_001_01_0013.dat 0 0
INPUT_FILE=anc45_001_01_0014.dat 0 0
INPUT_FILE=anc45_001_01_0015.dat 0 0
# DEM Files
INPUT_FILE=anc12_000_00_0000.dat  0 51926400
INPUT_FILE=anc12_000_00_0001.dat  0 51926400
# Geoid File
INPUT_FILE=anc13_001_01_0000.dat 00 51926400
INPUT_FILE=anc30_001_01_0000.dat 00 51926400
```

```
INPUT_FILE=anc31_001_01_0000.dat 00 51926400
INPUT_FILE=anc18_001_01_0000.dat 00 51926400
INPUT_FILE=anc35_001_01_0000.dat 00 51926400
INPUT_FILE=anc41_001_01_0000.dat 00 51926400
INPUT_FILE=ANC36_002_22_01005_2331_0_01_0000.DAT 51855160 51866758
INPUT_FILE=ANC36_002_22_01005_2333_0_01_0000.DAT 51866758 51878356
INPUT_FILE=anc17_001_01_0000.dat 51861600 51926400
INPUT_FILE=ANC01_000_20010823_120000_01_0000.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_120000_01_0001.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_120000_01_0002.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_120000_01_0003.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_120000_01_0004.DAT  51861600 51883200
INPUT_FILE=ANC01_000_20010823_180000_01_0000.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010823_180000_01_0001.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010823_180000_01_0002.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010823_180000_01_0003.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010823_180000_01_0004.DAT  51883200 51904800
INPUT_FILE=ANC01_000_20010824_000000_01_0000.DAT  51904800 51926400
INPUT_FILE=ANC01_000_20010824_000000_01_0001.DAT  51904800 51926400
INPUT_FILE=ANC01_000_20010824_000000_01_0002.DAT  51904800 51926400
INPUT_FILE=ANC01_000_20010824_000000_01_0003.DAT  51904800 51926400
INPUT_FILE=ANC01_000_20010824_000000_01_0004.DAT  51904800 51926400
#
INPUT_FILE=GLA02_002_22_01005_2331_0_01_0000.DAT 51855160 51866758
INPUT_FILE=GLA02_002_22_01005_2333_0_01_0000.DAT 51866758 51878356
#
OUTPUT_FILE=GLA07_002_22_01005_2331_0_01_0000.DAT 51855160 51866758
OUTPUT_FILE=GLA07_002_22_01005_2333_0_01_0000.DAT 51866758 51878356
OUTPUT_FILE=GLA08_002_22_01005_2331_0_01_0000.DAT 51855160 51866758
OUTPUT_FILE=GLA09_002_22_01005_2331_0_01_0000.DAT 51855160 51866758
OUTPUT_FILE=GLA10_002_22_01005_2333_0_01_0000.DAT 51866758 51878356
OUTPUT_FILE=GLA11_002_22_01005_2331_0_01_0000.DAT 51855160 51866758
#
OUTPUT_FILE=GLA07_002_22_01005_2331_0_01_0000.QAP 51855160 51866758
OUTPUT_FILE=GLA07_002_22_01005_2333_0_01_0000.QAP 51866758 51878356
OUTPUT_FILE=GLA08_002_22_01005_2331_0_01_0000.QAP 51855160 51866758
OUTPUT_FILE=GLA09_002_22_01005_2331_0_01_0000.QAP 51855160 51866758
OUTPUT_FILE=GLA10_002_22_01005_2333_0_01_0000.QAP 51866758 51878356
OUTPUT_FILE=GLA11_002_22_01005_2331_0_01_0000.QAP 51855160 51866758
OUTPUT_FILE=ANC06_001_20010823_180000_012342_00_GLAS_ATM.DAT 51861600 51883201
ATMOSPHERE_PROCESS=all
#
```

Section 10

GLAS_Reader

10.1 Function

GLAS_Reader reads nearly all GLAS product and ancillary files and prints the data contained within those files in human-readable format.

GLAS_Reader differs from other PGEs in that it has a rudimentary interface which will let the user specify input files from the command line. As with control file specifications, start and stop times are required along with the filename. The interface is invoked when running GLAS_Reader without a control file argument.

GLAS_Reader performs no processing, but permits specification of what type of data should be written to the text file. GLAS_Reader also supports input/output time selection.

Note: GLAS_Reader is not intended for production use.

10.2 Input Files

Table 10-1 lists the potential inputs files to GLAS_Reader. All or some of these files may be specified. Note, however, than GLA00 APID files may not be specified without also specifying a corresponding ANC29 file. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the non-specific files.

Table 10-1 GLAS_Reader Inputs

File Spec	Type	Source	Short Description
anc01*_????.dat	Dynamic Ancillary	met_util	Subsetted meteorological files. There is a separate ANC01 file per data type. All of the ANC01 files must be specified.
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error file.
anc07*_0001.dat	Static Ancillary	Science Team	GLAS global constants file.
anc07*_0002.dat	Static Ancillary	Science Team	GLAS waveform constants file.
anc07*_0003.dat	Static Ancillary	Science Team	GLAS elevation constants file.
anc07*_0004.dat	Static Ancillary	Science Team	GLAS atmosphere constants file.
anc07*_0005.dat	Static Ancillary	Science Team	GLAS L1A constants file.
anc08*.dat	Dynamic Ancillary	UTexas	Precision orbit file.
anc12*_0001.dat	Static Ancillary	Science Team	DEM mask file.

Table 10-1 GLAS_Reader Inputs

File Spec	Type	Source	Short Description
anc13*.dat	Static Ancillary	Science Team	Geoid file
anc16*.dat	Static Ancillary	Science Team	Ocean Tide file
anc17*.dat	Static Ancillary	Science Team	Load Tide file
anc18*.dat	Static Ancillary	Science Team	Standard Atmosphere file
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc27*.dat	Static Ancillary	Science Team	Regional mask files.
anc30*.dat	Static Ancillary	Science Team	Aerosol file
anc31*.dat	Static Ancillary	Science Team	Troposphere file
anc32*.dat	Dynamic Ancillary	GLAS_L0proc	Frequency board to GPS time correlation file.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.
anc36*.dat	Dynamic Ancillary	atm_anc	Atmosphere Calibration file.
anc45*.dat	Static Ancillary	Science Team	Metadata input files.
Control File	Control	ISIPS Operations	Control file.
gla00*.dat/ ANC29*.dat	Level-0 APID/ Dynamic Ancillary	EDOS/ GLAS_L0proc	GLAS Level-0 APID files and the requisite ANC29 index file.
gla*.dat	GLAS Product	GSAS	GLAS Product files.

10.2.1 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_Reader are described here.

The control file section delimiter for GLAS_Reader is:

```
=GLAS_Reader
```

Where appropriate, GLAS_Reader can print different types of data to the output file. The type of data printed is based on execution flags within the control file. The flags and the data types they control are defined in Table 10-2.

Table 10-2 GLAS_Reader Control Flags

Flag	Significance
READER_OPTION=PRODUCT	Indicates that GLAS_Reader should print product-type data (scaled integers).
READER_OPTION=ALGORITHM	Indicates that GLAS_Reader should print data converted as used in science algorithms.

Table 10-2 GLAS_Reader Control Flags

Flag	Significance
READER_OPTION=SCALE	Indicates that GLAS_Reader should print the scale factors.
READER_OPTION=ALL	Indicates that GLAS_Reader should print all of the data types listed above.

Combinations of the execution flags are allowed. For example, to print both algorithm and scale values, the control file would have the following entries:

```
READER_OPTION=ALGORITHM
READER_OPTION=SCALE
```

10.3 Output Files

GLAS_Reader requires no OUTPUT_FILE specifications in the control file. GLAS_Reader will create an output file for each type of input file requested. GLAS_Reader will strip the '.dat' from the specified input filename and replace it with a '.txt' extension. Time selection for the output files is based on the time specified with the input files.

A corresponding ANC29 file is required to process GLA00 APID files. When processing GLA00 APID files, GLAS_Reader writes all output to the ANC29 text file, instead of to individual APID files. The benefit of this is that the output is created in time-aligned fashion. Also note that specific APID files may be processed even though the ANC29 file was created with a superset of the selected APIDs.

10.4 Startup and Termination

This is an overview of the steps necessary to run GLAS_Reader.

10.4.1 Setup a Runtime Directory

The suggested method of running GLAS_Reader is to create a temporary directory and link all necessary files into it. For example, to setup a GLAS_Reader run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_Reader.
ln -s $GLAS_HOME/lib/*.
ln -s $GLAS_HOME/data/anc*.dat.
ln -s $DATA_DIR/*.dat.
```

10.4.2 Create a Control File

GLAS_Reader is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

cp $GLAS_HOME/data/cf_glas_reader.cf ./*control_file_name*
vi ./*control_file_name*

10.4.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

./GLAS_Reader *control_file_name*

10.4.4 Run-Time Statistics

Since GLAS_APID will not be used in the production environment, no runtime statistics have been recorded..

10.4.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

10.5 Error and Warning Messages

All GSAS error and warning messages are numerically listed in Appendix D.

10.6 Recovery Steps

If GLAS_Reader terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_Reader will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions

10.7 Sample GLAS_Reader Control File

This control file reads and writes 100 seconds of ANC29, ANC32, and GLA00 APIDs 17 and 19. It writes only Algorithm values to the output files. ANC32 data are written in anc32_002_20000101_000000_01_0000.txt. The ANC29 and GLA00 APID data are written to anc29_002_20000101_000000_01_0000.txt.

```
#
# This is the control file for the GLAS_Reader.
#
#----------Execution Information
#
=GLAS_Reader
EXEC_KEY=p3t1a
DATE_GENERATED=08 August 2001
OPERATOR=jlee
PGE_VERSION=2.2
#
#----------Dynamic ANC Files
#
# ANC29 and ANC32 files
#
INPUT_FILE=anc29_002_20000101_000000_01_0000.dat 0 100
INPUT_FILE=anc32_002_20000101_000000_01_0000.dat 0 100
#
# GLA00 files
#
INPUT_FILE=gla00_002_20000101_000000_01_0017.dat 0 100
INPUT_FILE=gla00_002_20000101_000000_01_0019.dat 0 100
#
#----------Execution Control
#
READER_OPTION=Algorithm
#
#----------End of Control File
#
```

Section 11

met_util

11.1 Function

This utility, met_util, creates ANC01 subset files (i.e. temperature, relative humidity, etc.) from a meteorological (ANC40) file. The ANC40 file is a standard GRIB file received from GSFC. The subset files are used as input for several of the GSAS PGEs. met_util reads the ANC07 file for error definitions.

met_util uses a subset of the standard GSAS control file with several modifications. The format of the met_util will be fully documented in a later subsection. met_util does not use an ANC06 file, has no processing control and no time selection capabilities.

met_util should be run before executing GLAS_Alt or GLAS_Atm (which require the subsetted ANC01 files) for the corresponding span of data.

11.2 Input Files

Table 11-1 lists the inputs files to met_util. The ANC40 file is the NCEP Global Analysis, a 1-by-1 degree gridded data set with sampling every 6 hours. Variables included are temperature, geopotential height, and relative humidity at standard upper atmospheric pressure levels. The ANC40 files are in the GRIB format, which is the WMO (World Meteorological Organization) standard for exchanging gridded binary data.All of these files must be specified and named according to GSAS file naming conventions.

Table 11-1 met_util Inputs

File Spec	Type	Source	Short Description
anc40*.dat	Dynamic Ancillary	GSFC	Standard MET file.
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error file.
anc07*_0006.dat	Static Ancillary	Science Team	GLAS utility constants file.
Control File	Control	ISIPS Operations	Control file.

11.2.1 Control File

The met_util control file is a subset of the standard GSAS control file. The basic format of this file is documented in Section 5 of this document. met_util does not use the common elements used in other GSAS control files.

The control file section delimiter for met_util is:

```
=met_util
```

The only keywords met_util uses are "INPUT_FILE", "OUTPUT_FILE", and the control file section delimiter. The "INPUT_FILE" and "OUTPUT_FILE" specifications do not have start time and stop times.

11.3 Output Files

Table 11-2 lists the specific subset output files created by met_util. The five files are an ASCII header file, a height file, a temperature file, a relative humidity file, and a precipitable water table file. The last four files are binary with no header.

Each record in the binary output file contains the global gridded values of the respective parameter at a particular pressure level. The first record is at the highest pressure level, and the last record is at the lowest pressure level. The ASCII header file lists the record number, location, date, pressure level, and parameter name (as read from the MET file). Refer to http://wesley.wwb.noaa.gov/wgrib.html for details

Output files must be named according to standard GSAS naming conventions.

Table 11-2 met_util Outputs

File Spec	Type	Destination	Short Description
anc01*_0000.dat	Dynamic Ancillary	GSAS PGEs/ Archive	Subsetted meteorological header file.
anc01*_0001.dat	Dynamic Ancillary	GSAS PGEs/ Archive	Subsetted meteorological precipitable water file.
anc01*_0002.dat	Dynamic Ancillary	GSAS PGEs/ Archive	Subsetted meteorological height file.
anc01*_0003.dat	Dynamic Ancillary	GSAS PGEs/ Archive	Subsetted meteorological relative humidity file.
anc01*_0004.dat	Dynamic Ancillary	GSAS PGEs/ Archive	Subsetted meteorological temperature file.

11.4 Startup and Termination

This is an overview of the steps necessary to run met_util. Specific operational procedures for each of the supported scenarios are provided in the ISIPS Operational Procedures Manual.

11.4.1 Setup a Runtime Directory

The suggested method of running met_util is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a met_util run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
```

```
ln -s $GLAS_HOME/bin/met_util .
ln -s $GLAS_HOME/bin/wgrib .
ln -s $GLAS_HOME/bin/SDMS_met_script .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc*.dat .
ln -s $DATA_DIR/*.dat .
```

11.4.2 Create a Control File

met_util is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

cp $GLAS_HOME/data/cf_met_util.ctl ./*control_file_name*
vi ./*control_file_name*

11.4.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

./met_util *control_file_name*

11.4.4 Run-Time Statistics

met_util subsets one file (6 hours) of meteorological data in less than a minute. This may vary due to hardware load and data dependencies.

11.4.5 Termination

The process will terminate automatically upon reaching the end of all input data. Standard output (stdout) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

11.5 Error and Warning Messages

Fatal errors are generated if there is a problem opening or reading a file. When an error occurs, an error message is printed to the standard output. It includes the error number, severity, process name where the error occurred, a short description, and the file name if the error occurred while accessing a file.

Errors possible from met_util with the severity and frequency include:

```
ERROR=-10002 No control file specified              3   1
ERROR=-10014 Unrecognized line in control file      3   1
ERROR=-10015 Unknown value in keyword/value pair    3   1
ERROR=-10017 I/O Error Reading Control File         3   1
ERROR=-10046 Error returned from script             3   1
```

```
ERROR=-10047 File naming conventions do not match    3   1
```

The error severity and frequency may be modified as described in the error section.

Explanations of these errors follow:

Table 11-3 Error Resolution

Error Type	Ensure that:
No control file specified	- control file name is typed correctly on command line
Reading File	- file is not available - file is not corrupt - regenerate if necessary
Unrecognized line in control file	- keywords have not been changed in control file - extraneous lines have not been added to control file
Unknown value in keyword/value pair	- input and output file names in control file conform to their correct naming conventions
File naming conventions do not match	- input and output file names in control file conform to their correct naming conventions - input and output file names match except for the last 7 characters
Error returned from script	- the script and executable are linked into the directory where the program is run - disk space is available

11.6 Recovery Steps

If met_util terminates with an error:

1) Review error and status messages on stdout to determine source or location of problem. Use Table 11-3 above for assistance.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of met_util will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions

11.7 Sample met_util Control File

This control file creates subsetted ANC01 files from the input ANC40 meteorological file.

```
= MET_UTIL
INPUT_FILE=anc40_000_20000101_120000_0001.dat
```

```
INPUT_FILE=anc07_000_00_0000.dat
INPUT_FILE=anc07_000_00_0006.dat
OUTPUT_FILE=anc01_000_20000101_120000_01_0000.dat
OUTPUT_FILE=anc01_000_20000101_120000_01_0001.dat
OUTPUT_FILE=anc01_000_20000101_120000_01_0002.dat
OUTPUT_FILE=anc01_000_20000101_120000_01_0003.dat
OUTPUT_FILE=anc01_000_20000101_120000_01_0004.dat
=
```

Section 12

reforbit_util

12.1 Function

This utility, reforbit_util, processes a given Reference Orbit file (ANC26) for all ascending equatorial crossings. Each ascending equatorial crossing will be given a track number. The first track west of Greenwich (or on Greenwich) will be assigned a Track number of 1 and its time will be determined. All consecutive tracks after that (in increasing time order) will be assigned numbers 2, 3, 4, and so on. All tracks that were to the right of Track 1, will be wrapped around the last track on the left and numbered accordingly.

This routine will also create an ECS NOSE file. The NOSE track file will be created from the along track locations. The routine will (starting from Track 1) scroll the along track locations at 161 second time increments. These increments correspond to 10 degree latitude increments. Polygons of approximately one km width in the longitude direction, and a ten degree length in the longitudinal direction will be created. These polygons will butt up against each other, with the end of the last polygon for track number n being the beginning of the first polygon for track number n+1. ECS requires track numbers to be sequential between the NOSE files. Therefore, a standalone routine, adjustNOSETrkNum, will be used to add an offset to the track number.

reforbit_util uses a modified version of the standard GSAS control file. The format of the reforbit_util control file will be fully documented in a later subsection. reforbit_util does not use an ANC06 file, has no processing control and no time selection capabilities.

reforbit_util should be run before executing GLAS_L1A, GLAS_Alt or GLAS_Atm (which require SDMS to provide orbit information) for the corresponding span of data.

12.2 Input Files

Table 12-1 lists the inputs files to reforbit_util. Refer to the “Interface Control Document Between I-SIPS/ISF and CSR” for a description of the Reference orbit file.All of these files must be specified and named according to GSAS file naming conventions.

Table 12-1 reforbit_util Inputs

File Spec	Type	Source	Short Description
anc26*.dat	Dynamic Ancillary	UTexas	Reference Orbit file.
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error file.

Table 12-1 reforbit_util Inputs

File Spec	Type	Source	Short Description
anc07*_0006.dat	Static Ancillary	Science Team	GLAS Utility constants file.
Control File	Control	ISIPS Operations	Control file.

12.2.1 Control File

The reforbit_util control file is a modified version of the standard GSAS control file. The basic format of this file is documented in Section 5 of this document. reforbit_util does not use the common elements used in other GSAS control files.

The control file section delimiter for reforbit_util is:

```
=reforbit_util
```

The only keywords reforbit_util uses are "INPUT_FILE", "INPUT_ERR", "OUTPUT_FILE1", "OUTPUT_FILE2", "OUTPUT_FILE3" and the control file section delimiter. These keyword specifications do not have start time and stop times. The keywords are used for the following designations:

Table 12-2 reforbit_util Keywords

Keyword	Value
INPUT_FILE	Input reference orbit filename.
INPUT_ERR	ANC07 error filename
OUTPUT_FILE1	output ANC22 reference orbit filename
OUTPUT_FILE2	SCF track file name(ANC 43)
OUTPUT_FILE3	NOSE track file name(ANC28)

12.3 Output Files

reforbit_util creates three files as output. The first file is an ASCII file (reference orbit track file), which contains the ascending node longitude and track number. The first record contains the average period of the tracks (in seconds), and the number of tracks in the reference orbit file. All subsequent records contain the longitude (in degrees E longitude), the track number, time in seconds relative to Track 1, the actual MJD time (in days), and the seconds of day.

The second file is in Fortran direct access format, with each record being 46412 bytes. This file is used by the SCF. It contains the 1-second along track locations for each track in the reference orbit file. Each 46412 byte record will contain 5801 latitudes (4-bytes each), 5801 longitudes (4-bytes each), and a 4-byte delta time. There will be one record for each track. The latitudes and longitudes are 1-second locations along each track from the start of the track to the end. The delta time indicates the time interval along the tracks (in this case 1 second).

The third file is an ASCII ECS NOSE track file. The format of the NOSE track is as follows:

A | 3 | track_1 | block_1

B | pt1_lat | pt1_lon | 0

B | pt2_lat | pt2_lon | 0

B | pt3_lat | pt3_lon | 0

B | pt4_lat | pt4_lon | 2

Where,

- Items in bold are fixed for ECS purposes,
- Longitudes (lon) values are expressed in decimal degrees (4 radix places), with values in the range [-180.0000° .. +180.0000°] [+E]
- Latitude (lat) values are expressed in decimal degrees (4 radix places), with values in the range [-90.0000° .. +90.0000°] [+E]
- Repeat A,B,B,B,B records for all blocks per track, and all tracks
- Points shall be ordered in a clockwise fashion, with no intersecting edges..

Table 12-3 reforbit_util Outputs

File Spec	Type	Destination	Short Description
anc22*.dat	Dynamic Ancillary	SDMS/Archive	Reference orbit file.
N/A	Dynamic Ancillary	SCF	SCF track file.
N/A	Dynamic Ancillary	ISIPS	NOSE track file

12.4 Startup and Termination

This is an overview of the steps necessary to run reforbit_util. Specific operational procedures for each of the supported scenarios are provided in the ISIPS Operational Procedures Manual.

12.4.1 Setup a Runtime Directory

The suggested method of running reforbit_util is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a reforbit_util run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/reforbit_util.
ln -s $GLAS_HOME/lib/*.
```

ln -s $GLAS_HOME/data/anc*.dat.
ln -s $DATA_DIR/*.dat.

12.4.2 Create a Control File

reforbit_util is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

cp $GLAS_HOME/data/cf_reforbit_util.ctl ./*control_file_name*
vi ./control_file_name

12.4.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

./reforbit_util *control_file_name*

12.4.4 Run-Time Statistics

reforbit_util processes one orbit file in less than a minute. This may vary due to hardware load and data dependencies.

12.4.5 Termination

The process will terminate automatically upon reaching the end of all input data. Standard output (stdout) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

12.5 Error and Warning Messages

Fatal errors are generated if there is a problem opening or reading a file. When an error occurs, an error message is printed to the standard output. It includes the error number, severity, process name where the error occurred, a short description, and the file name if the error occurred while accessing a file.

Errors possible from reforbit_util with the severity and frequency include:

```
ERROR=-10001 Error Opening File for Input          3   1
ERROR=-10002 No control file specified             3   1
ERROR=-10006 Error Opening File for Output         3   1
ERROR=-10014 Unrecognized line in control file     3   1
ERROR=-10017 I/O Error Reading Control File        3   1
ERROR=-10029 Error Returned From Script            3   1
ERROR=-50008 Value of cosine GT 1.0 in Geoloc      2   1
ERROR=-50009 Value of cosine Lt -1.0 in Geoloc     2   1
```

```
ERROR=-50010 Max Iterations exceeded in Geoloc       2   1
ERROR=-50013 Error reading 10 consec recs in POD file3   1
```

The error severity and frequency may be modified as described in the error section.

Explanations of these errors follow:

Table 12-4 Error Resolution

Error Type	Ensure that:
No control file specified	- control file name is typed correctly on command line
Reading File	- file is available - file is not corrupt - regenerate if necessary
Unrecognized line in control file	- keywords have not been changed in control file - extraneous lines have not been added to control file

12.6 Recovery Steps

If reforbit_util terminates with an error:

1) Review error and status messages on stdout to determine source or location of problem. Use Table 12-4 above for assistance.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of reforbit_util will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions

12.7 Sample reforbit_util Control File

This control file creates the ANC22 reference orbit file from the ANC26 input file.

```
= reforbit_util
INPUT_FILE=anc26_000_20000101_120000_0001.dat
INPUT_ERR= anc07_000_00_0000.dat
OUTPUT_FILE1=anc22_000_20000101_120000_0001.dat
OUTPUT_FILE2=anc43_20000101_120000_0001.dat
OUTPUT_FILE3=anc28_20000101_120000_0001.dat
=
```

Section 13

createGranule_util

13.1 Function

This utility, createGran_util, processes a given Predicted Orbit file and using a Reference Orbit ID file, creates 1/4 rev, 2 rev, and 14 rev granules. This utility will call the createGranule subroutine to create two scratch files (a rev file, and a granule file). The rev file will contain all the ascending equatorial crossings determined from the predicted orbit file. The granule file will contain the +/- 50 degree latitude crossings. The +/-50 degree latitude crossings will be designated by segment numbers. The segment numbers are defined in Table 13-1.

Table 13-1 Segment Description

Segment	Description
Segment 1	start of +50 degree latitude crossing (on the ascending portion of the track),
Segment 2	start of +50 degree latitude crossing (on the descending portion of the track),
Segment 3	start of a -50 degree latitude crossing (on the descending portion of the track),
Segment 4	start of a -50 degree latitude crossing (on the ascending portion of the track).

This utility will then check the operation mode selected (i.e., Predict or Reference). If the Reference mode is selected, then this utility will call the update_refTab subroutine to calculate the begin track, and time into begin track for the relevant reference orbit files. The Reference Orbit ID file (which is an input to this routine) will be updated with this new information. If the Predict mode is selected, then the calc_granules subroutine will be invoked. This routine will calculate the 1/4 rev, 2 rev, and 14 rev granule information on the basis of information contained in the Reference Orbit ID file and Predicted Orbit file. The calculated granule information will be written to user specified files. The information from the 1/4 rev granules will be used along with the appropriate track file to generate a SCF rev file. The SCF rev file will list the rev numbers, start times of each rev tracks as well as the reference orbit repeat ground track phase, reference orbit number and instance

createGran_util uses a modified version of the standard GSAS control file. The format of the createGran_util control file will be fully documented in a later subsection. createGran_util does not use an ANC06 file, has no processing control and no time selection capabilities.

createGran_util should be run before executing GLAS_L1A, GLAS_Alt or GLAS_Atm (which require SDMS to provide segment information) for the corresponding span of data.

13.2 Input Files

Table 13-2 lists the inputs files to createGran_util. Refer to the "Interface Control Document Between I-SIPS/ISF and CSR" for a description of the Predicted orbit file.All of these files must be specified and named according to GSAS file naming conventions.

Table 13-2 createGran_util Inputs

File Spec	Type	Source	Short Description
anc22*.dat	Dynamic Ancillary	UTexas	Predicted Orbit file.
anc42*.dat	Dynamic Ancillary	ISIPS	Reference orbit table.
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error file.
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
Control File	Control	ISIPS Operations	Control file.

13.2.1 Control File

The createGran_util control file is a modified version of the standard GSAS control file. The basic format of this file is documented in Section 5 of this document. createGran_util does not use the common elements used in other GSAS control files.

The control file section delimiter for createGran_util is:

```
=createGran_util
```

createGran_util keyword file specifications do not have start time and stop times. createGran_util uses the following keywords:

Table 13-3 createGran_util Keywords

Keyword	Value
IN_REFORBFILE	Reference orbit ID file name.
IN_PREDORBFILE	Predicted orbit table file name.
IN_ERRORFILE	Ancillary error file name.
IN_ANC25FILE	it is the GPS/UTC conversion file
LAST_TRACK	it is optional, but should be used when in transfer orbit mode
OUT_REFORBFILE	Reference orbit ID file name.
OUT_QRTREVFILE	quarter rev file name.
OUT_2REVFILE	two rev file name.
OUT_14REVFILE	fourteen rev file name
OUT_SCFREVFILE	SCF rev file name

Table 13-3 createGran_util Keywords (Continued)

Keyword	Value
PROC_MODE	Processing mode (i.e. PREDORB or REFORB)
START_TM	POD start time (J2000 secs)
STOP_TM	POD stop time (J2000 secs)

When running in REFORB mode, the IN_REFORBFILE, IN_PREDORBFILE, IN_ERRORFILE, IN_ANC25FILE, OUT_REFORBFILE, PROC_MODE, START_TM, and STOP_TM keywords are required. For the PREDORB mode, the IN_REFORBFILE, IN_PREDORBFILE, IN_ERRORFILE, IN_ANC25FILE, OUT_QRTREVFILE, OUT_2REVFILE, OUT_14REVFILE, OUT_SCFREVFILE, PROC_MODE, START_TM, and STOP_TM keywords are required. The LAST_TRACK keyword is optional, and is only used when in the transfer orbit mode. The value of the last track will be the last track that was processed by the predicted orbit file for the previous day.

13.3 Output Files

Four ASCII files are the primary output. The first file (called the quarter rev file) contains the quarter rev start time (in J2000 seconds), repeat ground track phase, reference orbit number, instance, product type (1 for quarter rev granule), cycle number, track number, and segment number for all the quarter rev granules determined from the predicted orbit file. The next ASCII file (the two rev file) contains the two rev start time (in J2000 seconds), repeat ground track phase, reference orbit number, instance, product type (2 for two rev granule), cycle number, track number, and segment number for all the two rev granules determined from the predicted orbit file. The third ASCII file (the fourteen rev file) contains the fourteen rev start time (in J2000 seconds), repeat ground track phase, reference orbit number, instance, product type (3 for two rev granule), cycle number, track number, and segment number for all the fourteen rev granules determined from the predicted orbit file. The fourth ASCII file (called the SCF rev file) contains the relative rev numbers, starting from 1 (during each execution), that were determined from the given predicted orbit file. It also contains the start time, repeat ground track phase, reference orbit number, instance, cycle number, and track number for all the equator crossings determined from the predicted orbit file.

Additionally, two ASCII scratch files will be created by the createGran_util Utility. The first file will be a rev file, which will contain the rev times and longitudes. Each row of this file will contain the time (in J2000 secs), latitude, and longitude of the ascending equatorial crossings. The other file will be a granule file, which will have the segment crossing times, and locations (latitudes and longitudes), as well as the segment numbers. Each line of the granule file will contain the start time (in J2000 secs), the latitude, longitude, and the segment number of a granule respectively. The first two rows will have a segment number of 10 and 11 respectively. These granules list the start and end times of the predicted orbit file respectively. For a two day predicted orbit file, the rev scratch file should be about 2 Kbytes, while a granule scratch file should be about 7Kbytes.

13.4 Startup and Termination

This is an overview of the steps necessary to run createGran_util. Specific operational procedures for each of the supported scenarios are provided in the ISIPS Operational Procedures Manual.

13.4.1 Setup a Runtime Directory

The suggested method of running createGran_util is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a createGran_util run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/createGran_util.
ln -s $GLAS_HOME/lib/*.
ln -s $GLAS_HOME/data/anc*.dat.
ln -s $DATA_DIR/*.dat.
```

13.4.2 Create a Control File

createGran_util is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

cp $GLAS_HOME/data/cf_creategran.ctl ./*control_file_name*
vi ./*control_file_name*

13.4.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

>./createGran_util *control_file_name*

13.4.4 Run-Time Statistics

createGran_util processes one orbit file in less than a minute. This may vary due to hardware load and data dependencies.

13.4.5 Termination

The process will terminate automatically upon reaching the end of all input data. Standard output (stdout) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

13.5 Error and Warning Messages

Fatal errors are generated if there is a problem opening or reading a file. When an error occurs, an error message is printed to the standard output. It includes the error number, severity, process name where the error occurred, a short description, and the file name if the error occurred while accessing a file.

Errors possible from createGran_util with the severity and frequency include:

```
ERROR=-10001 Error Opening File for Input           3   1
ERROR=-10002 No control file specified              3   1
ERROR=-10006 Error Opening File for Output          3   1
ERROR=-10014 Unrecognized line in control file      3   1
ERROR=-10017 I/O Error Reading Control File         3   1
ERROR=-10029 Error Returned From Script             3   1
ERROR=-50008 Value of cosine GT 1.0 in Geoloc       2   1
ERROR=-50009 Value of cosine Lt -1.0 in Geoloc      2   1
ERROR=-50010 Max Iterations exceeded in Geoloc      2   1
ERROR=-50013 Error reading 10 consec recs in POD file3  1
```

The error severity and frequency may be modified as described in the error section. Explanations of these errors follow:

Table 13-4 Error Resolution

Error Type	Ensure that:
No control file specified	- control file name is typed correctly on command line
Reading File	- file is available - file is not corrupt - regenerate if necessary
Unrecognized line in control file	- keywords have not been changed in control file - extraneous lines have not been added to control file

13.6 Recovery Steps

If createGran_util terminates with an error:

1) Review error and status messages on stdout to determine source or location of problem. Use Table 13-4 above for assistance.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of createGran_util will be available for test use. This is, however, more of a developer procedure than a user procedure.

13.7 Sample createGran_util Control File (REFORB)

This control file processes a reference orbit ID file with processing mode set to REFORB.

```
=CREATEGRAN_UTIL Control file createGran utility
IN_REFORBFILE=reforbIDfile_123
IN_PREDORBFILE=day3orb.da
IN_ANC25File=anc25_001_20021213_000000_01_0000.dat
IN_ERRORFILE=anc07_001_01_0000.dat
IN_ERRORFILE=anc07_001_01_0001.dat
OUT_REFORBFILE=reforbIDfile_123out
PROC_MODE=REFORB
START_TM= 173250.0
STOP_TM= 258945.0
=END of control file
```

13.8 Sample createGran_util Control File (PREDORB)

This control file processes a reference orbit ID file with processing mode set to PREDORB.

```
=CREATEGRAN_UTIL Control file createGran utility
IN_REFORBFILE=reforbIDfile_123out
IN_PREDORBFILE=day3orb.da
IN_ANC25File=anc25_001_20021213_000000_01_0000.dat
IN_ERRORFILE=anc07_001_01_0000.dat
IN_ERRORFILE=anc07_001_01_0001.dat
IN_ERRORFILE=anc07_001_01_0002.dat
IN_ERRORFILE=anc07_001_01_0003.dat
IN_ERRORFILE=anc07_001_01_0004.dat
IN_ERRORFILE=anc07_001_01_0005.dat
IN_ERRORFILE=anc07_001_01_0006.dat
OUT_QRTREVFILE=qrtrevfile_2
OUT_2REVFILE=tworevfile_2
OUT_14REVFILE=fteenrevfile_2
OUT_SCFREVFILE=SCFrevfile_2
PROC_MODE=PREDORB
START_TM= 173250.0
STOP_TM= 258945.0
=END of control file
```

Section 14
atm_anc

14.1 Function

Atm_anc reads a GLA02 product file and computes 532 nm and 1064 nm calibration coefficients for specified-time segments. The coefficients per segment are output to an ancillary (ANC36) file which is used in the level 1B atmosphere data processing. A second ancillary file (ANC44) contains the 532 and 1064 data for clouds that were detected above about 10 km. The file is intended to be used for the calibration of the 1064 channel, which will be done separately by the science team, outside of normal ISIPS processing.

This utility performs an intermediary step between the atmosphere level 1A and level 1B data processing. It reads the GLA02 product file output from the atmosphere level 1A processing and creates a file containing data used by the atmosphere level 1B data processing. This file contains 532 nm and 1064 nm calibration coefficients calculated over specified-time segments. This is achieved by summing 532 nm and 1064 nm lidar data over each second of the GLA02 product file for a specified time. Once the time has been reached, the data is averaged for the segment. Standard atmosphere and meteorological (MET) data is retrieved for each segment and average 532 nm and 1064 nm molecular backscatter values are computed for each segment. Averaged 532 nm ozone transmissions are also computed for each segment. From these data, the 532 nm and 1064 nm calibration coefficients are calculated for each segment. The calibration coefficients, molecular backscatter, and ozone transmissions for each segment are written to the ancillary output file.

atm_anc uses a modified version of the standard GSAS control file. The format of the atm_anc control file will be fully documented in a later subsection. atm_anc does not use an ANC06 file, has no processing control and no time selection capabilities.

atm_anc should be run after executing met_util and GLAS_L1A, but before executing GLAS_Atm (which requires the ANC36 and ANC44 files) for the corresponding span of data.

14.2 Input Files

Input files consist of the GLA02 product file, the standard atmosphere data file (ANC18), the ozone data file (ANC35), the MET data files (ANC01), and the ancillary (ANC07) files which contain constants used by the program and error numbers available to the program. Note that a full 12 hours of MET data (2 sets of files) must be specified. Table 14-1 lists the inputs files to atm_anc. All of these files must be specified and named according to GSAS file naming conventions.

Table 14-1 atm_anc Inputs

File Spec	Type	Source	Short Description
anc01*_??.dat	Dynamic Ancillary	met_util	Subsetted MET files. There is a separate MET file per MET data type. All of the MET files must be specified.
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error file.
anc07*_0002.dat	Static Ancillary	Science Team	GLAS atmosphere constants file.
anc18*.dat	Static Ancillary	Science Team	Standard Atmosphere file
anc35*.dat	Static Ancillary	Science Team	Ozone file
Control File	Control	ISIPS Operations	Control file.
gla02*.dat	L1A Product	GLAS_L1A	GLAS L1A Atmosphere product file.

14.2.1 Control File

The atm_anc control file is a modified version of the standard GSAS control file. The basic format of this file is documented in Section 5 of this document. atm_anc does not use the common elements used in other GSAS control files.

The control file section delimiter for atm_anc is:

```
=atm_anc
```

The only keywords atm_anc uses are "INPUT_FILE", "OUTPUT_FILE" and the control file section delimiter.

14.3 Output Files

The software creates ANC36 and ANC44 output files.

Table 14-2 atm_anc Outputs

File Spec	Type	Destination	Short Description
anc36*.dat	Dynamic Ancillary	GLAS_Atm/Archive	Atmosphere Calibration file.
anc44*.dat	Dynamic Ancillary	Science Team	Atm 1064 Cirrus CAL File

14.3.1 ANC36

The ANC36 output file contains 23 values per segment:

- first latitude
- first longitude
- last latitude

- last longitude
- first record index
- first time (J2000 sec)
- last time (J2000 sec)
- 532 calibration coefficient at the high calibration height
- 532 calibration coefficient at the low calibration height
- 1064 calibration coefficient at the low calibration height
- average 532 lidar value at the high calibration height
- average 532 lidar value at the low calibration height
- average 1064 lidar value at the low calibration height
- average 532 molecular backscatter value at the high calibration height
- average 532 molecular backscatter value at the low calibration height
- average 1064 molecular backscatter value at the low calibration height
- average 532 ozone transmission value at the high calibration height
- average 532 ozone transmission value at the low calibration height
- average 532 background
- high calibration height (m)
- high calibration band width (m)
- low calibration height (m)
- low calibration band width (m)

The number of segments is the first number in the file.

An example of the output file follows:

```
2
   first lat     first lon     last lat      last lon
   rec_index     first_time    last_time      ccof_ghi      ccof_glo      ccof_irlo
    lid_ghi         lid_glo        lid_irlo       mol_ghi        mol_glo        mol_irlo
    oz_ghi        oz_glo     g_bkgrd         cht_hi         hi_wid        cht_lo
lo_wid
 ------------------------------------------------------------------------------------
------------------------
   50.1268    354.2534    84.9712     306.0022
  30010.           3001.         3589.  3.513E+13  4.695E+13  5.684E+02
 7.952E+05  1.274E+07  1.021E-05  2.281E-08   2.795E-07   1.797E-08
  9.921E-01  9.712E-01  5.509E+00    30000.00      2000.00     14848.00    2000.00
                 36010.          3601.         4189.  3.924E+13  5.412E+13  9.802E+02
 8.847E+05  1.265E+07  1.516E-05  2.264E-08   2.413E-07   1.547E-08
  9.961E-01  9.686E-01  5.510E+00    30000.00      2000.00     14848.00    2000.00
```

The output file is meant to be used by the GSAS atmosphere level 1B data processing routines. These routines read the ANC36 file and interpolate the 532 nm and 1064 nm calibration coefficients per segment to once per second. These coefficients are then used to compute the 532 nm and 1064 nm backscatter cross section profiles at once per second for use in the atmosphere level 2 data processing.

The ANC36 file must be named according to standard GSAS naming conventions

14.3.2 ANC44

For each cloud above 'GD_CIRRUS_HT' the ANC44 file contains a header line followed by 'GI_CIRRUS_BINS' bins of 532 and 1064 NRB (GLA02) data, averaged to 1 second. The header line contains the time, latitude, longitude, cloud height, bin number of the first bin printed out and the number of bins printed out. There can be many clouds for a given granule, so the size of each ANC44 file can be considerable (1 MB or so).

An example of the ANC44 output for one cloud follows:

```
51890013.     50.127    354.585  13285.400   360    6
  4.37720000D+06   3.50340000D-05
  9.44040000D+06   8.62500000D-05
  8.54600000D+06  -5.03000000D-06
  4.14280000D+06   8.40680000D-05
  7.54640000D+06   7.29500000D-05
  3.51900000D+06   5.51380000D-05
```

14.4 Startup and Termination

This is an overview of the steps necessary to run atm_anc. Specific operational procedures for each of the supported scenarios are provided in the ISIPS Operational Procedures Manual.

14.4.1 Setup a Runtime Directory

The suggested method of running atm_anc is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a atm_anc run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
>mkdir $TEMP_DIR
>cd $TEMP_DIR
>ln -s $GLAS_HOME/bin/atm_anc.
>ln -s $GLAS_HOME/lib/*.
>ln -s $GLAS_HOME/data/anc*.dat.
>ln -s $DATA_DIR/*.dat.
```

14.4.2 Create a Control File

atm_anc is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

```
>cd $TEMP_DIR
>cp $GLAS_HOME/data/control_template ./control_file_name
>vi ./control_file_name
```

14.4.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

```
>cd $TEMP_DIR
>./atm_anc control_file_name
```

14.4.4 Run-Time Statistics

atm_anc processes one orbit file in approximately x minutes. This may vary due to hardware load and data dependencies.

14.4.5 Termination

The process will terminate automatically upon reaching the end of all input data. Standard output (stdout) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

14.5 Error and Warning Messages

Fatal errors are generated if there is a problem opening or reading a file. When an error occurs, an error message is printed to the standard output. It includes the error number, severity, process name where the error occurred, a short description, and the file name if the error occurred while accessing a file.

Errors possible from atm_anc with the severity and frequency include:

```
ERROR=-10001 Error Opening File for Input: 31

ERROR=-10002 No control file specified31

ERROR=-10006 Error Opening File for Output31

ERROR=-10008 Error Reading File31

ERROR=-10009 Error Writing File31

ERROR=-10014 Unrecognized line in control file31

ERROR=-10015 Unknown value in keyword/value pair 3 1

ERROR=-10017 I/O Error Reading Control File31

ERROR=-10025 Error Reading Standard Atmosphere file header
3 1

ERROR=-10026 Error Reading Standard Atmosphere file data
3 1

ERROR=-40001 DEM out-of-bounds 150
```

```
ERROR=-40024 Invalid value written to ancillary file 11
ERROR=-40025 Array index is greater than size allowed31
```

The error severity and frequency may be modified as described in the error section.

Explanations of these errors follow:

Table 14-3 Error Resolution

Error Type	Ensure that:
No control file specified	- control file name is typed correctly on command line
Opening File	- file is available - control file reflects the correct file name - file name does not exceed 60 characters
Reading File	- file is available - file is not corrupt - regenerate if necessary
Writing File	- file is available - disk space is available
Unrecognized line in control file	- keywords have not been changed in control file - extraneous lines have not been added to control file
Unknown value in keyword/ value pair	- input and output file names in control file conform to their correct naming conventions
Array index is greater than size allowed	- the dimension of the array is sufficient for the data it needs to contain - the amount of data is not too large for the array

14.6 Recovery Steps

If atm_anc terminates with an error:

1) Review error and status messages on stdout to determine source or location of problem. Use Table 14-3 above for assistance.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of atm_anc will be available for test use. This is, however, more of a developer procedure than a user procedure.

14.7 Sample atm_anc Control File

```
=ATM_ANC
INPUT_FILE=GLA02_002_22_01005_2331_0_01_0000.DAT
```

```
INPUT_FILE=ANC01_000_20010823_120000_01_0000.DAT
INPUT_FILE=ANC01_000_20010823_120000_01_0001.DAT
INPUT_FILE=ANC01_000_20010823_120000_01_0002.DAT
INPUT_FILE=ANC01_000_20010823_120000_01_0003.DAT
INPUT_FILE=ANC01_000_20010823_120000_01_0004.DAT
INPUT_FILE=ANC01_000_20010823_180000_01_0000.DAT
INPUT_FILE=ANC01_000_20010823_180000_01_0001.DAT
INPUT_FILE=ANC01_000_20010823_180000_01_0002.DAT
INPUT_FILE=ANC01_000_20010823_180000_01_0003.DAT
INPUT_FILE=ANC01_000_20010823_180000_01_0004.DAT
INPUT_FILE=ANC01_000_20010824_000000_01_0000.DAT
INPUT_FILE=ANC01_000_20010824_000000_01_0001.DAT
INPUT_FILE=ANC01_000_20010824_000000_01_0002.DAT
INPUT_FILE=ANC01_000_20010824_000000_01_0003.DAT
INPUT_FILE=ANC01_000_20010824_000000_01_0004.DAT
INPUT_FILE=anc07_001_01_0000.dat
INPUT_FILE=anc07_001_01_0001.dat
INPUT_FILE=anc07_001_01_0002.dat
INPUT_FILE=anc07_001_01_0003.dat
INPUT_FILE=anc07_001_01_0006.dat
INPUT_FILE=anc07_001_01_0005.dat
INPUT_FILE=anc07_002_01_0004.dat
INPUT_FILE=anc18_001_01_0000.dat
INPUT_FILE=anc35_001_01_0000.dat
OUTPUT_FILE=ANC36_002_22_01005_2331_0_01_0000.DAT
OUTPUT_FILE=ANC44_002_22_01005_2331_0_01_0000.DAT
```

Section 15
GLAS_Meta

15.1 Function

GLAS_Meta reads ANC45 and ANC46 metadata input files and, using the parsed information, reads the header records of GLAS product and ancillary files to create ECS-compliant inventory-level metadata files. GLAS_Meta may be either run immediately after specific product files have been generated or just before sending batches of product/ancillary files for distribution. GLAS_Meta allows for multiple-file/multi-granule input and output files.

15.2 Input Files

Table 15-1 lists the inputs to GLAS_Meta. Files which are specific to GLAS_Meta are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the those files not specific to GLAS_Meta.

Table 15-1 GLAS_Meta Inputs

File Spec	Type	Source	Short Description
gla*.dat	GLAS Products	GSAS	GLAS product files.
anc45*.dat	Static Ancillary	Science Team	Product metadata input files.
anc04*.dat	Dynamic Ancillary	UTexas	IERS Polar Motion and Earth Rotation Data File.
anc46*_0004.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC04.
anc08*.dat	Dynamic Ancillary	UTexas	Precision Orbit file.
anc46*_0008.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC08.
anc09*.dat	Dynamic Ancillary	UTexas	Precision Attitude file.
anc46*_0009.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC09.
anc20*.dat	Dynamic Ancillary	UTexas	Predicted orbit file.
anc46*_0020.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC20.
anc22*.dat	Dynamic Ancillary	ISIPS	Track file.
anc46*_0022.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC22.
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.

Table 15-1 GLAS_Meta Inputs (Continued)

File Spec	Type	Source	Short Description
anc46*_0025.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC25.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.
anc46*_0033.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC33.
anc37*.dat	Dynamic Ancillary	UTEXAS	Spacecraft CG file.
anc46*_0037.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC37.
anc39*.dat	Dynamic Ancillary	UTEXAS	GPS file.
anc46*_0039.dat	Static Ancillary	Science Team	Ancillary metadata input file for ANC39.
anc07*_0001.dat	Static Ancillary	Science Team	GLAS global constants file.
Control File	Control	ISIPS Operations	Control file.

15.2.1 ANC45 Product Metadata Input Files

ANC45 Metadata input files are text-based templates created from ESDTs (Earth Science Data Types). The GLAS team collaborated with NSIDC to create ESDTs for each product and each ancillary product. Within the ESDT are collection-level and inventory-level metadata. Only the inventory-level metadata is present in the ANC45 files.

Each ANC45 filetype corresponds to the similarly numbered GLA product file. The files are parsed and empty fields are filled in from values present in the appropriate product header records.

Note that a single ANC04*_04 file is used for GLA04 processing and only the GLA04*_06.dat file is required to satisfy the GLA04 metadata processing requirements.

15.2.2 ANC46 Ancillary Metadata Input Files

ANC46 Metadata input files are text-based templates created from ESDTs (Earth Science Data Types). The GLAS team collaborated with NSIDC to create ESDTs for each product and each ancillary product. Within the ESDT are collection-level and inventory-level metadata. Only the inventory-level metadata is present in the ANC46 files.

Each ANC46 filetype corresponds to the similarly numbered ancillary file. The files are parsed and empty fields are filled in from values related to the target ancillary file.

15.2.3 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_Meta are described in this section.

The control file section delimiter for GLAS_Meta is:

```
=GLAS_Meta
```

Since GLAS_Meta has no requirement for execution scenarios, there are no unique keywords for the GLAS_Meta control file. GLAS_Meta will perform all functions based on the presence of input and output files within the control file.

15.3 Output Files

Table 15-2 lists the outputs created by GLAS_Meta. Files which are specific to GLAS_Meta are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the files which are not specific to GLAS_Meta.

Table 15-2 GLAS_Meta Outputs

File Spec	Type	Destination	Short Description
gla*.met	Metadata	ECS	ECS-compliant metadata inventory files.
anc04*.met	Metadata	ECS	IERS Polar Motion and Earth Rotation metadata File.
anc08*.met	Metadata	ECS	Precision Orbit metadata file.
anc09*.met	Metadata	ECS	Precision Attitude metadata file.
anc20*.met	Metadata	ECS	Predicted orbit metadata file.
anc22*.met	Metadata	ECS	Track metadata file.
anc25*.met	Metadata	ECS	GPS/UTC conversion metadata file.
anc33*.mett	Metadata	ECS	UTC time conversion metadata file.
anc37*.met	Metadata	ECS	Spacecraft CG metadata file.
anc39*.met	Metadata	ECS	GPS metadata file.
anc06*.txt	Dynamic Ancillary	ISIPS Operations	Standard metadata/processing log file.

15.3.1 GLA Metadata Files

GLAS metadata files are required to enable ECS to ingest GLAS products into their system. One metadata file must be generated for each GLAS product file delivered to ECS. Each GLAS product file type has a corresponding metadata file type (eg: gla05*.met corresponds to gla05*.dat). The exception is GLA04. Whereas GLA04 is a multi-file granule, a single metadata file is sufficient for each GLA04 granule. The gla04*_06.dat file should be used as input and gla04*06.met as output when processing GLA04 metadata information.

15.4 Processing and Reprocessing Scenarios

GLAS_Meta supports only a full-processing scenario. All processing is based on the presence of input and output files within the control file.

Additionally, time-selected processing is not supported. The start and stop times on the INPUT_FILE and OUTPUT_FILE control file entries are ignored for processing.

15.5 Startup and Termination

This is an overview of the steps necessary to run GLAS_Meta.

15.5.1 Setup a Runtime Directory

The suggested method of running GLAS_Meta is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a GLAS_Meta run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_Meta .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc07*.dat .
ln -s $GLAS_HOME/data/anc45*.dat .
ln -s $DATA_DIR/*.dat .
```

15.5.2 Create a Control File

GLAS_Meta is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

```
cp $GLAS_HOME/data/cf_glas_meta.ctl ./control_file_name
vi ./control_file_name
```

15.5.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

```
./GLAS_Meta control_file_name
```

15.5.4 Run-Time Statistics

GLAS_Meta has minimal processing requirements since it only parses the product header records.

15.5.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

15.6 Error and Warning Messages

All GSAS error and warning messages are listed numerically in the Appendix D.

15.7 Recovery Steps

If GLAS_Meta terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem. Refer to errors listed in Appendix D for assistance in determining the problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_Meta will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions.

15.8 Sample GLAS_Meta Control File

```
=GLAS_Meta
#
#---------Execution Information
#
EXEC_KEY=p3t1
PGE_VERSION=V2.2
DATE_GENERATED=11 January 2002
OPERATOR=jlee
#
#----------Static ANC Files
#
# Input ANC07 Files : 00=error, 01=global
#
INPUT_FILE=anc07_001_01_0000.dat 0000000 1000000
INPUT_FILE=anc07_001_01_0001.dat 0000000 1000000
#
# Input Metadata Templates
#
INPUT_FILE=anc45_001_01_0001.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0002.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0003.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0004.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0005.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0006.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0007.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0008.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0009.dat 0000000 100000000
```

```
INPUT_FILE=anc45_001_01_0010.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0011.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0012.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0013.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0014.dat 0000000 100000000
INPUT_FILE=anc45_001_01_0015.dat 0000000 100000000
#
# Input gla files
#
INPUT_FILE=GLA01_002_22_01005_2332_1_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA02_002_22_01005_2331_0_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA04_002_22_01005_2331_0_01_0006.DAT 0 52000000 1 1 0
INPUT_FILE=GLA05_002_22_01005_2332_1_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA06_002_22_01005_2332_1_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA07_002_22_01005_2331_0_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA08_002_22_01005_2331_0_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA09_002_22_01005_2331_0_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA10_002_22_01005_2333_0_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA11_002_22_01005_2331_0_01_0003 DAT 0 52000000 1 1 0
INPUT_FILE=GLA12_002_22_01005_2325_0_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA13_002_22_01005_2325_0_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA14_002_22_01005_2325_0_01_0000.DAT 0 52000000 1 1 0
INPUT_FILE=GLA15_002_22_01005_2325_0_01_0000.DAT 0 52000000 1 1 0
#
#----------Output ANC Files
#
# Output ANC06
#
OUTPUT_FILE=anc06_glas_meta.dat 0000000 1000000
#
#----------Output GLA Files
#
OUTPUT_FILE=GLA01_002_22_01005_2332_1_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA02_002_22_01005_2331_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA05_002_22_01005_2332_1_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA04_002_22_01005_2331_0_01_0006.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA06_002_22_01005_2332_1_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA07_002_22_01005_2331_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA08_002_22_01005_2331_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA09_002_22_01005_2331_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA10_002_22_01005_2333_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA11_002_22_01005_2331_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA12_002_22_01005_2325_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA13_002_22_01005_2325_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA14_002_22_01005_2325_0_01_0000.MET 0 52000000 1 1 0
OUTPUT_FILE=GLA15_002_22_01005_2325_0_01_0000.MET 0 52000000 1 1 0
#
#----------End of Control File
#
```

Section 16

GLAS_GPS

16.1 Function

GLAS_GPS is a Fortran 90 wrapper written around the JPL-supplied RINEX (Receiver Independent Exchange Format) processing software. GLAS_GPS reads the APID 1088 GPS data and products (via RINEX) and ANC39 ASCII file containing processed GPS information. GLAS_GPS allows for multi-granule input and output files.

Note: the functionality of GLAS_GPS has been depreciated. An agreement was reached whereas UTCSR will process the APID 1088 files themselves. GLAS_GPS will not be used in ISIPS production processing and is not being maintained.

16.2 Input Files

Table 16-1 lists the inputs to GLAS_GPS. Files which are specific to GLAS_GPS are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the those files not specific to GLAS_GPS.

Table 16-1 GLAS_GPS Inputs

File Spec	Type	Source	Short Description
gla00*_1088.dat	Level-0 APID	EDOS	GLAS GPS APID file.
anc07*_01.dat	Static Ancillary	Science Team	GLAS global constants file.
Control File	Control	ISIPS Operations	Control file.

16.2.1 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_GPS are described in this section.

The control file section delimiter for GLAS_GPS is:

```
=GLAS_GPS
```

Since GLAS_GPS has no requirement for execution scenarios, there are no unique keywords for the GLAS_GPS control file. GLAS_GPS will perform all functions based on the presence of input and output files within the control file.

16.3 Output Files

Table 16-2 lists the outputs created by GLAS_GPS. Files which are specific to GLAS_GPS are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the files which are not specific to GLAS_GPS.

Table 16-2 GLAS_GPS Outputs

File Spec	Type	Destination	Short Description
anc39*.datt	Dynamic Ancillary	UTCSR	Text-based output from the RINEX processing software.
anc06*.dat	Dynamic Ancillary	ISIPS Operations	Standard metadata/processing log file.

16.3.1 ANC39 RINEX Output Files

For more information on RINEX – See the document "RINEX: The Receiver Independent Exchange Format Version 2".

16.4 Processing and Reprocessing Scenarios

GLAS_GPS supports only a full-processing scenario. All processing is based on the presence of input and output files within the control file.

Additionally, time-selected processing is not supported. The start and stop times on the INPUT_FILE and OUTPUT_FILE control file entries are ignored for processing.

16.5 Startup and Termination

This is an overview of the steps necessary to run GLAS_GPS. Specific operational procedures will be provided in the ISIPS Operational Procedures Manual.

16.5.1 Setup a Runtime Directory

The suggested method of running GLAS_GPS is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a GLAS_GPS run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_GPS .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc07*.dat .
ln -s $DATA_DIR/*.dat .
```

16.5.2 Create a Control File

GLAS_GPS is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

```
cp $GLAS_HOME/data/cf_GLAS_GPS.ctl ./control_file_name
vi ./control_file_name
```

16.5.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

```
./GLAS_GPS control_file_name
```

16.5.4 Run-Time Statistics

Since GLAS_GPS will not be used in the production environment, no runtime statistics have been recorded..

16.5.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

16.6 Error and Warning Messages

All GSAS error and warning messages are listed numerically in the Appendix D.

16.7 Recovery Steps

If GLAS_GPS terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem. Refer to errors listed in Appendix D for assistance in determining the problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_GPS will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions.

16.8 Sample GLAS_GPS Control File

```
=GLAS_GPS
#
#---------Execution Information
#
EXEC_KEY=p3t1
PGE_VERSION=V2.2
DATE_GENERATED=11 January 2002
OPERATOR=jlee
```

```
#
#----------Static ANC Files
#
# Input ANC07 Files : 00=error, 01=global
#
INPUT_FILE=anc07_001_01_00.dat 0000000 1000000
INPUT_FILE=anc07_001_01_01.dat 0000000 1000000
#
# Input GPS APID
#
INPUT_FILE=gla00_008_20020406_010000_01_1088.dat 51861600 51883201
#
#----------Output ANC Files
#
OUTPUT_FILE=anc06_GLAS_GPS.dat 0000000 1000000
OUTPUT_FILE=anc39_008_20020406_010000_01_1088.dat 51861600 51883201
#
#----------End of Control File
#
```

Section 17
GLAS_APID

17.1 Function

GLAS_APID is a modified form of GLAS_Reader which will read GLA00 APID files and create spreadsheet-friendly, column-based text files. Since GLAS_APID does not attempt to time-align the APIDS, (unlike GLAS_Reader) the ANC29 file is not required (but supported). GLAS_APID currently only processes ANC29, ANC32, and APID19 but its functionality will be expanded to include most APIDs of interest.

Like GLAS_Reader, GLAS_APID has a rudimentary interface which will let the user specify input files from the command line. As with control file specifications, start and stop times are required along with the filename. The interface is invoked when running GLAS_APID without a control file argument.

GLAS_APID performs no processing, but permits specification of what type of data should be written to the text file. GLAS_APID also supports input/output time selection.

Note: GLAS_APID is not intended for production use.

17.2 Input Files

Table 17-1 lists the inputs to GLAS_APID. Files which are specific to GLAS_APID are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the those files not specific to GLAS_APID.

Table 17-1 GLAS_APID Inputs

File Spec	Type	Source	Short Description
anc29*.dat	Dynamic Ancillary	GLAS_L0proc	Index file correlating APID times.
anc32*.dat	Dynamic Ancillary	GLAS_L0proc	Frequency board to GPS time correlation file.
gla00*_19.dat	Level-0 APID	EDOS	GLAS Anc Sci APID file.
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error file.
anc07*_0001.dat	Static Ancillary	Science Team	GLAS global constants file.
Control File	Control	ISIPS Operations	Control file.

17.2.1 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_APID are described in this section.

The control file section delimiter for GLAS_APID is:

```
=GLAS_APID
```

Where appropriate, GLAS_APID can output the first element or all elements of pre-record data to the output file. The default is to output only the first array element. To designate that full-rate output should be created, use the flag defined in Table 17-2.

Table 17-2 GLAS_APID Control Flag

Flag	Significance
APID_RATE=FULLRATE	Indicates that GLAS_APID should write full-rate array values.

17.3 Output Files

GLAS_APID requires no OUTPUT_FILE specifications in the control file. GLAS_APID will create one or more output files for each type of input file requested. GLAS_APID will append extensions onto the base filename to create output files. Time selection for the output files is based on the time specified with the input files.

Table 17-3 lists the outputs created by GLAS_APID. Files which are specific to GLAS_APID are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the files which are not specific to GLAS_APID.

Table 17-3 GLAS_APID Outputs

File Spec	Type	Destination	Short Description
anc29*.dat.txt	text	Science Team	anc29 data in text format.
anc32*.dat.txt	text	Science Team	anc32 data in text format.
gla00*_19.dat_time.txt	text	Science Team	time-related parameters from apid19 in text format.
gla00*_19.dat_ad.txt	text	Science Team	AD-related parameters from apid19 in text format.
gla00*_19.dat_pc.txt	text	Science Team	PC-related parameters from apid19 in text format.
gla00*_19.dat_cd.txt	text	Science Team	CD-related parameters from apid19 in text format.
gla00*_19.dat_gps.txt	text	Science Team	GPS-related parameters from apid19 in text format.
gla00*_19.dat_ct.txt	text	Science Team	CT-related parameters from apid19 in text format.

17.3.1 APID19 Text Output Files

One of the ways to make the text files more spreadsheet-friendly is to make the number of columns in the output file manageable. Therefore, the APID19 text output is broken up into multiple text files grouped by related parameters.

17.4 Processing and Reprocessing Scenarios

GLAS_APID supports only a full-processing scenario. All processing is based on the presence of input and output files within the control file.

Time-selected processing is supported. The start and stop times on the INPUT_FILE and OUTPUT_FILE control file entries delimit the range of data output.

17.5 Startup and Termination

This is an overview of the steps necessary to run GLAS_APID.

17.5.1 Setup a Runtime Directory

The suggested method of running GLAS_APID is to create a temporary directory and link all necessary files into it. For example, to setup a GLAS_APID run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_APID .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc07*.dat .
ln -s $DATA_DIR/*.dat .
```

17.5.2 Create a Control File

GLAS_APID is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

```
cp $GLAS_HOME/data/cf_GLAS_APID.ctl ./control_file_name
vi ./control_file_name
```

17.5.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

```
./GLAS_APID control_file_name
```

17.5.4 Run-Time Statistics

Since GLAS_APID will not be used in the production environment, no runtime statistics have been recorded..

17.5.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

17.6 Error and Warning Messages

All GSAS error and warning messages are listed numerically in the Appendix D.

17.7 Recovery Steps

If GLAS_APID terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem. Refer to errors listed in Appendix D for assistance in determining the problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_APID will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions.

17.8 Sample GLAS_APID Control File

This control file reads and writes 100 seconds of ANC29, ANC32, and GLA00 APID 19.

```
#
# This is the control file for the GLAS_APID.
#
#----------Execution Information
#
=GLAS_APID
EXEC_KEY=p3t1a
DATE_GENERATED=08 August 2001
OPERATOR=jlee
PGE_VERSION=2.2
#
#----------Dynamic ANC Files
#
# ANC29 and ANC32 files
#
INPUT_FILE=anc29_002_20000101_000000_01_00.dat 0 100
INPUT_FILE=anc32_002_20000101_000000_01_00.dat 0 100
#
# GLA00 files
```

```
#
INPUT_FILE=gla00_002_20000101_000000_01_19.dat 0 100
#
#----------Execution Control
#
READER_OPTION=Algorithm
#
#----------End of Control File
#
```

Section 18

GLAS_Tick

18.1 Function

GLAS_Tick reads ANC09, ANC32 and (optionally) GLA03 input files to create ANC50_00 and ANC50_01 output files. The ANC50_00 contains merged ANC09/ANC32 information which is written at a GPS update event. The ANC50_01 contains 6 hour statistics for oscillator and engineering data. GLAS_Tick is a full-fledged PGE and allows for multiple-file/multi-granule input and output files. Please reference the "Construction of the Lookup Table of GLAS Clock Oscillator Frequencies" memo by C. Field, X. Sun, and D. Hancock for requirements specifications.

18.2 Input Files

Table 18-1 lists the inputs to GLAS_Tick. Files which are specific to GLAS_Tick are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the those files not specific to GLAS_Tick.

Table 18-1 GLAS_Tick Inputs

File Spec	Type	Source	Short Description
gla03.dat	GLAS Products	GSAS	GLAS ENG product files.
anc09*.dat	Dynamic Ancillary	UTexas	Precision Attitude file.
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc32*.dat	Dynamic Ancillary	Science Team	GPS time correction file.
anc33*.dat	Dynamic Ancillary	Science Team	UTC time conversion file.
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error constants file.
anc07*_0001.dat	Static Ancillary	Science Team	GLAS global constants file.
anc07*_0005.dat	Static Ancillary	Science Team	GLAS L1A constants file.
Control File	Control	ISIPS Operations	Control file.

18.2.1 Control File

The control file format and common elements are documented in Section 5 of this document. Elements specific to GLAS_Tick are described in this section.

The control file section delimiter for GLAS_Tick is:

```
=GLAS_Tick
```

As a full-fledged PGE, control file sanity checking is provided. The following conditions are checked and the PGE will error-out if a condition is violated:

- There must be exactly 1 anc25 input file.
- There must be exactly 1 anc33 input file.
- There must be anc07 input files.
- There must be at least one ANC32 input file.
- There must be at least one ANC09 input file.
- There must be at least one ANC50_0000 and ANC50_0001 output file.
- There must be the same number of ANC50_0000 and ANC50_0001 output files.

Since GLAS_Tick has no requirement for execution scenarios, there are no unique keywords for the GLAS_Tick control file. GLAS_Tick will perform all functions based on the presence of input and output files within the control file.

Time controls on input/output files specification lines are enforced.

18.3 Output Files

Table 18-2 lists the outputs created by GLAS_Tick. Files which are specific to GLAS_Tick are documented in this section. See the appropriate section of this document or the GLAS Data Products Specifications Volumes for details regarding the files which are not specific to GLAS_Tick.

Table 18-2 GLAS_Tick Outputs

File Spec	Type	Destination	Short Description
anc50*_0000.txt	Text file	ISF/Science Team	Merged ANC09/ANC32 file.
anc50*_0001.txt	Text file	ISF/Science Team	Frequency trend file.
anc06*.txt	Dynamic Ancillary	ISIPS Operations	Standard metadata/processing log file.

18.3.1 anc50_0000.txt File

The anc50_0000 file is a merged ANC09/ANC32 file. The file contains multiple lines of tab-separated fields with descriptive headers. The files is designed to import cleanly into MS Excel. A line is written each time a GPS update occurs (once per 10 seconds). For parameters whose rate is greater than once per second, only the first value is written. Parameters are written in the following format (where A1 represents a TAB character):

```
I16,2(A1,F20.4),A1,F20.9,8(A1,F20.4),2(A1,I16),7(A1,F20.4),A1,A3
```

18.3.2 anc50_0001.txt Files

The anc50_0001 file contains statistics computed from parameters in the ANC09/GLA03 files. The file contains multiple lines of tab-separated fields with descriptive headers. The files is designed to import cleanly into MS Excel. Each line contains statistics for (nominally) 6 hours of data. When computing min/mean/max statistics for parameters whose rate is greater than one per second, all values are used in the computation. Additionally, those parameters which are component-sensitive use values consistent with the active instrument. Depending upon the presence of GLA03 input (see next section) parameters are written in one of the two following format (where A1 represents a TAB character):

(with GLA03 inputs)

```
2(I16,A1),6(F20.4,A1),I6,5(A1,F20.14),33(A1,F20.4)
```

(without GLA03 inputs)

```
2(I16,A1),6(F20.4,A1),I6,5(A1,F20.14)
```

18.4 Processing and Reprocessing Scenarios

GLAS_Tick supports two different processing scenarios, dependent on the presence of GLA03 input files. Simply, if GLA03 input files are present, GLA03 statistics are computed and written. If GLA03 inputs are not present, no GLA03 statistics are computed or written. These scenarios are based entirely on the presence of GLA03 input files within the control file.

18.5 Startup and Termination

This is an overview of the steps necessary to run GLAS_Tick.

18.5.1 Setup a Runtime Directory

The suggested method of running GLAS_Tick is to emulate what the SDMS will do. The SDMS will create a temporary directory and link all necessary files into it. For example, to setup a GLAS_Tick run, one would perform the following steps (Designate TEMP_DIR as the temporary directory and DATA_DIR as the location in which input data for this job has been staged):

```
mkdir $TEMP_DIR
cd $TEMP_DIR
ln -s $GLAS_HOME/bin/GLAS_Tick .
ln -s $GLAS_HOME/lib/* .
ln -s $GLAS_HOME/data/anc07*.dat .
ln -s $GLAS_HOME/data/anc25*.dat .
ln -s $GLAS_HOME/data/anc33*.dat .
ln -s $DATA_DIR/*.dat .
```

18.5.2 Create a Control File

GLAS_Tick is designed to take the name of a control file as a command-line argument. The suggested method of creating a control file is by copying a template from the $GLAS_HOME/data directory and modifying it with desired input/output filenames and data processing options.

```
cp $GLAS_HOME/data/cf_GLAS_Tick.ctl ./control_file_name
vi ./control_file_name
```

18.5.3 Run the PGE

Error and status messages will be displayed on screen (stdout) and recorded in the ANC06 file.

```
./GLAS_Tick control_file_name
```

18.5.4 Run-Time Statistics

GLAS_Tick has minimal processing requirements since it only parses the product header records.

18.5.5 Termination

The process will terminate automatically upon reaching the end of all input data. The log/metadata file (ANC06*) must be examined to determine runtime success. Additionally, the process will return a result code to the operating system which may be used to programatically determine success or failure (0=success, 3=fatal error).

18.6 Error and Warning Messages

All GSAS error and warning messages are listed numerically in the Appendix D.

18.7 Recovery Steps

If GLAS_Tick terminates with an error:

1) Review error and status messages in the ANC06 file or on stdout to determine source or location of problem. Refer to errors listed in Appendix D for assistance in determining the problem.
2) Correct the problem
3) Remove previous output files
4) Re-run the software

In case of a problematic error, which cannot be easily diagnosed, debug versions of GLAS_Tick will be available for test use. This is, however, more of a developer procedure than a user procedure. Software users should contact the GSAS Development Team for more instructions.

18.8 Sample GLAS_Tick Control File

```
=GLAS_Tick
#----------Start of Control File
#
# This is an control file automatically generated by
# create_tick_ctl.sh.
#
#---------Execution Information
#
EXEC_KEY=auto
DATE_GENERATED=Thu Jan  8 18:25:19 EST 2004
OPERATOR=ACC_TEST
```

```
PGE_VERSION=ACC_TEST
#
#----------Input Static ANC Files
#
# Input ANC07
#
INPUT_FILE=anc07_013_01_0000.dat  0 1  1 1 0
INPUT_FILE=anc07_013_01_0001.dat  0 1  1 1 0
INPUT_FILE=anc07_013_01_0005.dat  0 1  1 1 0
INPUT_FILE=anc25_001_20021213_000000_01_0000.dat  0 1  1 1 0
INPUT_FILE=ANC33_001_20031223_193523_01_0000.DAT 0 1
#
#----------Input/Output Files
#
INPUT_FILE=ANC32_012_20030925_180000_01_0001.DAT 0 1
INPUT_FILE=ANC32_012_20030926_000000_01_0001.DAT 0 1
INPUT_FILE=ANC32_012_20030926_060000_01_0001.DAT 0 1
INPUT_FILE=ANC32_012_20030926_120000_01_0001.DAT 0 1
INPUT_FILE=ANC32_012_20030926_180000_02_0001.DAT 0 1
INPUT_FILE=ANC32_012_20030927_000000_02_0001.DAT 0 1

INPUT_FILE=GLA03_013_1102_028_0091_0_01_0001.DAT 117799207.0 117810806.0
INPUT_FILE=GLA03_013_1102_028_0093_0_01_0001.DAT 117810807.0 117822407.0
INPUT_FILE=GLA03_013_1102_028_0095_0_01_0001.DAT 117822408.0 117834006.0
INPUT_FILE=GLA03_013_1102_028_0097_0_01_0001.DAT 117834007.0 117845607.0
INPUT_FILE=GLA03_013_1102_028_0099_0_01_0001.DAT 117845608.0 117857206.0
INPUT_FILE=GLA03_013_1102_028_0101_0_01_0001.DAT 117857207.0 117868815.0
INPUT_FILE=GLA03_013_1102_028_0103_0_01_0001.DAT 117868816.0 117880414.0
INPUT_FILE=GLA03_013_1102_028_0105_0_01_0001.DAT 117880415.0 117892014.0
INPUT_FILE=GLA03_013_1102_028_0107_0_01_0001.DAT 117892015.0 117903615.0


INPUT_FILE=ANC09_005_20030925_235947_01_0001.DAT 117806400.0 117892800.0 1 1 0
OUTPUT_FILE=ANC50_005_20030925_235947_01_0000.txt 117806400.0 117892800.0 1 1 0
OUTPUT_FILE=ANC50_005_20030925_235947_01_0001.txt 117806400.0 117892800.0 1 1 0
OUTPUT_FILE=ANC06_tick_005_20030925_235947_01_0001.DAT   1 1 0
#
#----------Execution Control
#
# (none)
#
#----------End of GLAS_Tick Section---------------------------
#
```

Section 19

QABrowse

19.1 Introduction

Program Name: QABrowse

Language: Interactive Data Language (IDL)

Function: Read quality assurance files for the data products and output image files with plots of the data that can be examined with a web browser.

19.2 Overview of Purpose and Functions

Purpose: Generate images that can be browsed over the web to let users determine whether they want to look at the data in the granules.

Function: Read QAP files and output image files.

19.3 Startup and Termination

To run the program:

1) Change to a runtime directory
2) Generate the control file.
3) The following three directories should be in your path:

```
/<GLAS_HOME>/idl/qa_browse/browse
/<GLAS_HOME>/idl/qa_browse/read
/<GLAS_HOME>/idl/qa_browse/util
```

where GLAS_HOME is as defined in previous sections of this document. These can be added to your IDL path with the command

IDL> **!path = '/<GLAS_HOME>/idl/qa_browse/browse:' $**

+ '/<GLAS_HOME>/idl/qa_browse/read:' $

+'/<GLAS_HOME>/idl/qa_browse/util:' !path

issued after you enter IDL and BEFORE you run the program. Here you need to substitute the value of GLAS_HOME in place of each <GLAS_HOME> since IDL does not know what this environment variable is. For example, if

GLAS_HOME = /data/myworkingdir,

the actual statement would be

IDL> **!path = '/data/myworkingdir/idl/qa_browse/browse:' $**

+ '/data/myworkingdir/idl/qa_browse/read:' $

+ '/data/myworkingdir/idl/qa_browse/util:' !path

4) Either

a. [i] Enter IDL and at the IDL prompt type

IDL> **qabrowse, controlFileName, outputFormat**

where outputFormat is 'png', 'ps', or 'x' (case-insensitive).

The default output format is PNG.

[ii] Exit from IDL.

OR

b. at the shell prompt or in a script, enter

$> runbrowse controlFileName

5. Move the output files (if you generated any) whereever they belong, if necessary.

19.4 Functions and Their Operation

The main functions of this program are:

- Read the control file
- Read the QAP files
- Generate QA browse images.

19.5 Input Arguments

19.5.1 Control File

This program requires the name of a control file as a command-line argument. The program reads input from the control file to determine control parameters. A sample control file is provided in the code directory (sample.txt) and is shown below.

19.6 Input Files

19.6.1 Control File

The program reads through the file until it finds a line containing

```
=QABROWSE
```

and nothing else. At that point, it reads and interprets all lines until it finds another line with an = sign in column 1, at which point the program stops reading, closes the control file, and begins processing.

Only lines of the form

```
KEY=Value
```

are evaluated, and they may be in any order.

The following KEY=Value pairs have meaning. Except for INPUT_FILE, only the last occurrence of each one is used.

Table 19-1 Description of Control File Contents

ENVIRONMENT=Value	**OPTIONAL**. Value = ISIPS or SCF, depending on where the product files were generated. The SCF version generates png files with different file names, and is not as strict with respect to the expected lat/lon range of the data. SCF will work for the ISIPS files but the plots may not have nice limits in some cases. **Example**: `ENVIRONMENT=ISIPS` **Default**: ISIPS
VERSION=Value	**REQUIRED** (not optional) as of 2003 Apr. 11. As of 2003 Apr. 11, this is the version of the GSAS software used to generate the QAP file. Note this is a real number, not a string, so it cannot handle subreleases such as 3.4.5. For previous versions of the program, use the following: VERSION = 1 is obsolete and should not be used. VERSION = 2 is for files produced by version 3.0 or earlier of the GSAS software (MOSS 10 testing). VERSION = 3 is for files produced after version 3.0 of the GSAS software. It differs from VERSION=2 in that GLA06 and 12-15 QAP files were written as one record per along-track record, followed by a separator record consisting of one along-track record filled with invalids, followed by one summary record written as a single record. The along-track records contain averages of the trop and tide corrections, and the summary record contains histograms of these quantities plus number of elements in each histogram and lower limit of the first histogram bin and upper limit of the last histogram bin. VERSION=4 differs from VERSION=3 in that global summaries of the range parameters were added to the summary records in the elevation qap files (GLA06,12-15). **Example**: `VERSION=3.6`

Table 19-1 Description of Control File Contents (Continued)

OUTPUT_FORMAT=Value	**OPTIONAL**. Value should be a string. This parameter specifies the format for the output plots. **Allowed values**:PNG (file output in png format) X (screen plots on an x-window system) PS (file output in Postscript format) **Example**: `OUTPUT_FORMAT=PNG` **Default**: PNG

Table 19-1 Description of Control File Contents (Continued)

OUTPUT_DIRECTORY= Value	**OPTIONAL**. Value is a string containing the name of the directory in which the output files (if any) are to be stored. If the output directory line is present and the value does not end in a / (on Unix systems) or \ (on Windows systems), one is added. **Example**: `OUTPUT_DIRECTORY=/home/jack/` **Default**: current directory.

INPUT_FILE=Value	There are two types of input files.
	
	Type 1: REQUIRED. Value should be a string containing the name of the QAP file to be processed. It is assumed that the file name is of the form "path/qapxx..." (without the quotes), where xx is a 2-digit number between 01 and 15. "path/" is the full path to the file, and may be left off if the file is in the working directory. "path" and "qap" may be any mix of upper and lower case characters but must correspond exactly to the actual path and file names. INPUT_FILE may appear any number of times, in which case all QAP files listed are processed. At least one INPUT_FILE is required. If there are no lines of this form, the program will assume there is no data to be processed and will quietly stop.
	
	Example:
	
	`INPUT_FILE=/XOVER/disk4/jack/`
	`GLA06_009_1102_004_0065_3_01_0001.qap`
	
	Type 2: REQUIRED for GLA03. Processing of GLA03 files requires an ANC07_001_01_0005 file.
	
	Example:
	
	`INPUT_FILE= ANC07_001_01_0005.DAT`

Table 19-2 Sample Control File

```
=QABROWSE
VERSION=3.6
ENVIRONMENT=GSAS

# First QAP file to be processed
INPUT_FILE=/home/jack/
GLA03_003_20030925_164545_01_0000.QAP
# A second QAP file to be processed
INPUT_FILE=/home/jack/glas/
GLA06_012_20031005_180404_01_0000.QAP

INPUT_FILE=/home/jack/anc07_001_01_0005.dat
OUTPUT_DIRECTORY=/home/jack
OUTPUT_FORMAT=png
=END
```

19.6.2 QAP File

One or more QAP files are required as input.

19.6.3 ANC File

For GLA03 processing, the ANC07 0005 file is required to get the parameter limits.

19.7 Output

19.7.1 GLA01

For GLA01, the program generates four plots for each input QAP file.

File 1: global summary

Files 2-3: bars showing min, max, average, std dev, and number of measurements for a variety of parameters

File 4: time histories

19.7.2 GLA02

For GLA02, the program generates three plots for each input QAP file.

File 1: summary information, groundtrack map of integrated return

File 2: 532 nm data

File 3: 1064 nm data

19.7.3 GLA03

For GLA03, the program generates a global summary file with a list of % missing data, followed by a list of out-of-bound parameters and how many times each was out of bounds. This list may extend to several pages.

The remaining pages contain bar plots showing min/max/average/std dev/number of points for a variety of parameters.

19.7.4 GLA04

For GLA04, the program generates eight plots. Plots 1-4 are global summary data and plots 5-8 contain along-track averages.

Plot 1: summary LPA and LRS images and Granule statistics.

Plots 2 and 3: granule statistics for a number of LRS, BST, and IST parameters.

Plot 4: histograms of a number of BST, IST, and LRS parameters.

Plot 5: along-track averages of LPA and Gyro parameters.

Plot 6: along-track averages of LRS parameters.

Plot 7: along-track averages of Solar array, Attitude, and BST parameters.

Plot 8: along-track averages of several IST parameters.

19.7.5 GLA05

For GLA05, the program generates either 13 or 16 plots for each input QAP file. Fewer files are generated for equatorial regions because these do not have any ice sheet regions. There are three different types of plots:

1) Global summary data. Only one of these per input QAP file.
2) Histograms. One file for each surface type, and one for all surface types combined.
3) Groundtrack maps of along-track data. Two files for each surface type, and one for all surface types combined.

19.7.6 GLA06

For GLA06, the program generates two to four plots for each input QAP file. Fewer files are generated for earlier versions of GLA06 QAP files.

Plot 1: summary data, histograms, and groundtrack maps of along-track data

Plot 2 and 3: groundtrack maps of along-track data

Plot 4: bars showing min, max, average, std dev, and number of measurements for the range adjustment parameters.

19.7.7 GLA07

For GLA07, the program generates one global summary plot.

19.7.8 GLA08

For GLA08, the program generates one global summary plot.

19.7.9 GLA09

For GLA09, the program generates one global summary plot.

19.7.10 GLA10

For GLA10, the program generates one global summary plot.

19.7.11 GLA011

For GLA11, the program generates one global summary plot.

19.7.12 GLA12

For GLA12, the program generates two to four plots for each input QAP file. Fewer files are generated for earlier versions of GLA12 QAP files.

Plot 1: summary data, histograms, and groundtrack maps of along-track data

Plots 2 and 3: groundtrack maps of along-track data

Plot 4: bars showing min, max, average, std dev, and number of measurements for the range adjustment parameters.

19.7.13 GLA13

For GLA13, the program generates two to four plots for each input QAP file. Fewer files are generated for earlier versions of GLA13 QAP files.

Plot 1: summary data, histograms, and groundtrack maps of along-track data

Plots 2 and 3: groundtrack maps of along-track data

Plot 4: bars showing min, max, average, std dev, and number of measurements for the range adjustment parameters.

19.7.14 GLA14

For GLA14, the program generates two to four plots for each input QAP file. Fewer files are generated for earlier versions of GLA14 QAP files.

Plot 1: summary data, histograms, and groundtrack maps of along-track data

Plots 2 and 3: groundtrack maps of along-track data

Plot 4: bars showing min, max, average, std dev, and number of measurements for the range adjustment parameters.

19.7.15 GLA015

For GLA15, the program generates two to four plots for each input QAP file. Fewer files are generated for earlier versions of GLA15 QAP files.

Plot 1: summary data, histograms, and groundtrack maps of along-track data

Plots 2 and 3: groundtrack maps of along-track data

Plot 4: bars showing min, max, average, std dev, and number of measurements for the range adjustment parameters.

19.8 Error and Warning Messages

All warning and error messages generated by the program are self-explanatory. Any error messages generated by IDL should be reported except for floating underflows. These error messages are generated by the IDL internals and there does not appear to be any way to avoid getting them at least some of the time.

If the program generates the png files successfully, IDL exits with exit code QA_NoError = 0.

If the program ends without generating all png files, IDL exits with a nonzero exit code. The exit codes are defined in file util/qa_exitcodes.pro.

19.9 Recovery Steps

Exit IDL, correct the error in the control file, and start over.

Section 20
QAPG

20.1 Introduction

Program Name: QAPG, from **QAP** Generator

Language: Fortran 90

Function: Read product files (granules) and generate QAP files

20.2 Overview of Purpose and Functions

Purpose: Read product files (granules) and generate QAP files

Function: Read product files (granules) and generate QAP files

20.3 Startup and Termination

To run the program:

1) Change to a runtime directory
2) Generate a control file.
3) At the shell prompt or in a script, enter

   $> **qapg controlFileName**

20.4 Functions and Their Operation

The main functions of this program are:

- Read the control file
- Read the product file
- Generate QAP file

20.5 Input Arguments

20.5.1 Control File

This program requires the name of a control file as a command-line argument. The program reads input from the control file to determine control parameters. A sample control file is provided in the code directory (sample.txt) and is shown below.

20.6 Input Files

20.6.1 Control File

The program reads through the file until it finds a line containing

```
=PROD_VER
```

and nothing else. The = should be in column 1 and there should be no space between it and the P. At that point, the program reads and interprets all lines until it finds another line with an = sign in column 1, at which point the program stops reading, closes the control file, and begins processing.

Only lines of the form

```
KEY=Value
```

are evaluated, and they may be in any order.

20.6.2 Product File(s)

One (or six for GLA04) product file is required as input.

20.6.3 ANC Files

ANC07 0000, 0003, 0004, and 0006 are required.

20.7 Output

QAPG generates one QAP file each time it is run successfully.

20.8 Error and Warning Messages

There are some of these in the portion of the code written by Kristine Barbieri.

20.9 Recovery Steps

Table 20-1 Description of Control File Contents

INPUT_FILE=Value	There are two types of input files.
	
	Type 1: REQUIRED. Value should be a string containing the name of the product file to be processed.
	
	In general, it is assumed that the file name is of the form "GLAxx..." (without the quotes), where xx is a 2-digit number between 01 and 15, and "GLA" may be any mix of upper and lower case characters but must correspond exactly to the actual path and file names. GLA04 file names have special requirements.
	
	For GLA01 - GLA03 and GLA05 - GLA15, INPUT_FILE may appear only once.
	
	For GLA04, INPUT_FILE must appear six times, and the name must be of the form
	
	GLAxx...000Y.zzz,
	
	where Y = 1,2,3,4,5, and 6. The rest of the file name must follow the rules for the other products.
	
	Y designates the type of GLA04 product file, as follows:
	
	1: LPA
	
	2: LRS
	
	3: Gyro
	
	4: IST
	
	5: BST
	
	6: Spacecraft
	
	Example:
	
	`INPUT_FILE=`
	`GLA01_012_1102_002_0013_4_01_0001.DAT`
	
	Type 2: REQUIRED. ANC07 files. The following ANC files are required
	
	ANC07 0000
	
	ANC07 0003
	
	ANC07 0004
	
	ANC07 0006
	
	Example: INPUT_FILE= ANC07_001_01_0000.DAT

Table 20-1 Description of Control File Contents

OUTPUT_FILE=Value	**REQUIRED**. Value should be a string containing the name of the output QAP file. **Example**: GLA01_012_1102_002_0013_4_01_0001.QAP

Table 20-2 Sample Control File

```
=PROD_VER
INPUT_FILE=GLA01_012_1102_002_0013_4_01_0001.DAT
INPUT_FILE=anc07_001_01_0000.dat
INPUT_FILE=anc07_001_01_0003.dat
INPUT_FILE=anc07_001_01_0004.dat
INPUT_FILE=anc07_001_01_0006.dat
OUTPUT_FILE=gla01_012_1102_002_0013_4_01_Integ_0001.qap
=END
```

Section 21

QAPCompare

21.1 Introduction

Program Name: QAPCompare.

Language: Interactive Data Language (IDL)

Function: Generate Quality Assurance parameters and flags based on data in the QAP files.

21.2 Overview of Purpose and Functions

Purpose: Generate product-specific QA parameters to allow data quality assessment, and QA flags for inclusion in the metadata.

Functions:

1) Generate Metadata flags and parameters from a QA file.
2) Compare the QA file generated during production of the product granule with the QA file generated from the product granule.

NOTE: It is possible to run the program without the second QA file, in which case only the first function is performed.

NOTE: Flag output is not implemented for all products.

21.3 Startup and Termination

To run the program:

1) Change to a runtime directory
2) Generate the control file.
3) The following three directories should be in your path:

   /<GLAS_HOME>/idl/qa_browse/compare

   /<GLAS_HOME>/idl/qa_browse/read

   /<GLAS_HOME>/idl/qa_browse/util

where GLAS_HOME is previously defined in this document. These can be added to your IDL path with the command

```
IDL> !path = '/<GLAS_HOME>/idl/qa_browse/compare:' $
    + '/<GLAS_HOME>/idl/qa_browse/read:' $
    + '/<GLAS_HOME>/idl/qa_browse/util:' !path
```

issued after you enter IDL and BEFORE you run the program. Here you need to substitute the value of GLAS_HOME in place of each <GLAS_HOME> since IDL does not know what this environment variable is. For example, if

GLAS_HOME = /data/myworkingdir,

the actual statement would be

IDL> **!path = '/data/myworkingdir/idl/qa_browse/compare:' $**

+ '/data/myworkingdir/idl/qa_browse/read:' $

+ '/data/myworkingdir/idl/qa_browse/util:' !path

4) Either

a) [i] Enter IDL and at the IDL prompt type

IDL> **qapcompare, controlFileName**

[ii] Exit from IDL.

OR

b) at the shell prompt or in a script, enter

$> **runcompare controlFileName**

5) Move the output files (if you generated any) to wherever they belong, if necessary.

21.4 Functions and Their Operation

The main functions of this program are:

- Read the control file
- Read the QAP file(s)
- Generate the AUTOQA parameters and flags based on data in the QAP file that was made along with the product granule.
- Generate the OPERQA parameters and flags if one or more are defined for the product.

Optionally, generate the VAV file giving detailed comparison of two QAP files.

21.5 Input Arguments

21.5.1 Control File

This program requires the name of a control file as a command-line argument. The program reads input from the control file to determine control parameters. A sample control file is provided in the code directory (sample.txt), and a description of the file is given below.

21.6 Input Files

21.6.1 Control File

The program reads through the control file until it finds a line containing

=QAPCOMPARE

and nothing else. The = sign must be in column 1 and there may be no spaces in the line. It reads and interprets all lines following this until it finds another line with an = sign in column 1, at which point the program stops reading, closes the control file, and begins processing. Blank lines and lines with an octothorpe (#) in column 1 are ignored. All other lines are expected to have the form

KEY=Value,

and they may be in any order. Spaces around the = sign are ignored, so

KEY =   Value

is treated the same as

KEY=Value

An example file accompanies the distribution, in file sample.txt.

Table 21-1 KEY=Value Entries for QAPCompare

KEY=Value Entry	Description
QAP_FILE=Value	**REQUIRED**. Value should be a string containing the name of the QAP file generated along with the product granule. It is assumed that the file name is of the form "path/GGGxx..." (without the quotes), where xx is a 2-digit number between 01 and 15. "path/" is the full path to the file, and may be left off if the file is in the working directory, and "GGG" is exactly 3 characters. "path" and "GGG" may be any mix of upper and lower case characters, which must correspond exactly to the actual path and file names. **EXAMPLE**: QAP_FILE=GLA01_012_1102_002_0013_4_01_0001.QAP
QVV_FILE=Value	**OPTIONAL** as of 2003 October. Value should be a string containing the name of the QAP file generated from the product granule. It's format is identical to the format of QAP_FILE. **EXAMPLE**: QVV_FILE=GLA01_012_1102_002_0013_4_01_0011.QAP

Table 21-1 KEY=Value Entries for QAPCompare

OUTPUT_VAV=Value	**OPTIONAL**. Name of the VAV file to be written by the program. Ignored if QVV_FILE is not present. May include a path. **EXAMPLE**: OUTPUT_VAV = GLA01.VAV
OUTPUT_QAPFLAGS=Value	**REQUIRED**. Value should be a string containing the name of the QAPFlags file to be written by the program. May include a path. **EXAMPLE**: OUTPUT_QAPFLAGS = GLA01.QAP-Flags

Table 21-1 KEY=Value Entries for QAPCompare

DOALONGTRACK=Value	**OPTIONAL**. Value should be **TRUE** or **FALSE**. TRUE ==> compare the along-track records. FALSE ==> do NOT compare the along-track records. In general, DOALONGTRACK should be set to FALSE for GLA12-15, TRUE for all other products. **EXAMPLE**: DOALONGTRACK = FALSE
VERSION=Value	**REQUIRED**. As of 2003 Apr. 11, this is the version of the GSAS software used to generate the QAP file. REQUIRED (not optional) as of 2003 Apr. 11. **EXAMPLE**: VERSION=3.6 **NOTE**: this is a real number, not a string, so it cannot handle subreleases such as 3.4.5. For previous versions of the program, use the following: VERSION = 1 is obsolete and should not be used. VERSION = 2 is for files produced by version 3.0 or earlier of the GSAS software (MOSS 10 testing). VERSION = 3 is for files produced after version 3.0 of the GSAS software. It differs from VERSION=2 in that GLA06 and 12-15 QAP files were written as one record per along-track record, followed by a separator record consisting of one along-track record filled with invalids, followed by one summary record written as a single record. The along-track records contain averages of the trop and tide corrections, and the summary record contains histograms of these quantities plus number of elements in each histogram and lower limit of the first histogram bin and upper limit of the last histogram bin. VERSION=4 differs from VERSION=3 in that global summaries of the range parameters were added to the summary records in the elevation qap files (GLA06,12-15).

21.7 Output

21.7.1 VAV file

An ASCII file containing a detailed comparison of the two QAP files. The file contains a list of all components of the QAP file summary record and along-track records (if DOALONGTRACK=True), with statistics for each, and a comparison of the two QAP files. The only difference from product to product is that the specific components change.

21.7.2 QAPFlags file

ASCII file with the QA flags and parameters.

21.8 Understanding the Output

21.8.1 QAPFlags File

The QAPFLAGS file contains a list of product-specific flags and parameters in KEYWORD=Value format. The specific keywords vary from product to product. Sample output files for each product follow.

Table 21-2 Sample QAPFLAGS Files

KEY=Value Entry	Description
GLA01.QAPFlags	AUTOQA_UTCTime_PctMissing = 0.061842919 AUTOQA_Waveform_PctMissing = 0.061842919 AUTOQA_Waveform_PctOOB = N/A AUTOQA_UTCTime_FLAG = PASS AUTOQA_Waveforms_FLAG = PASS
GLA02.QAPFlags	AUTOQA_UTCTime_PctMissing = 0.0086199466 AUTOQA_PCProf_PctMissing = 0.0086199466 AUTOQA_CDProf_PctMissing = 0.0086199466 AUTOQA_PCProf_PctOOB = N/A AUTOQA_CDProf_PctOOB = N/A AUTOQA_UTCTime_FLAG = PASS AUTOQA_PCProf_FLAG = PASS AUTOQA_CDProf_FLAG = PASS
GLA03.QAPFlags	AUTOQA_UTCTime_PctMissing = 0.0086199466 AUTOQA_Temp_PctOOB = N/A AUTOQA_Voltage_PctOOB = N/A AUTOQA_UTCTime_FLAG = PASS AUTOQA_Temperature_FLAG = PASS AUTOQA_Voltage_FLAG = PASS

Table 21-2 Sample QAPFLAGS Files

GLA04.QAPFlags	AUTOQA_BST_PctMissing = 1.7976931e+308 AUTOQA_Gyro_PctMissing = 1.7976931e+308 AUTOQA_IST_PctMissing = 1.7976931e+308 AUTOQA_LPA_PctMissing = 1.7976931e+308 AUTOQA_LRS_PctMissing = 1.7976931e+308 AUTOQA_PRAP_PctMissing = 1.7976931e+308 AUTOQA_LPA_FLAG = FAIL AUTOQA_PRAP_FLAG = FAIL AUTOQA_LRS_FLAG = N/A AUTOQA_BST_FLAG = N/A AUTOQA_IST_FLAG = N/A AUTOQA_Gyro_FLAG = N/A
GLA05.QAPFlags	AUTOQA_RANGE_FLAG = PASS RANGE_PCTMISSING = 34.7800 ICESHEET_NVALIDMEASUREMENTS = N/A SEAICE_NVALIDMEASUREMENTS = N/A LAND_NVALIDMEASUREMENTS = N/A OCEAN_NVALIDMEASUREMENTS = N/A
GLA06.QAPFlags	AUTOQA_SURFELEVATION_FLAG = FAIL SURFELEVATION_PctOOB = N/A SURFROUGHNESS_PctOOB = N/A SURFREFLECTANCE_PctOOB = N/A SURFSLOPE_PctOOB = N/A
GLA07.QAPFlags	AUTOQA_ = N/A
GLA08.QAPFlags	AUTOQA_ = N/A
GLA09.QAPFlags	AUTOQA_ = N/A
GLA10.QAPFlags	AUTOQA_ = N/A
GLA11.QAPFlags	AUTOQA_ = N/A
GLA12.QAPFlags	AUTOQA_SURFELEVATION_FLAG = PASS AUTOQA_SURFROUGHNESS_FLAG = PASS AUTOQA_SURFREFLECTANCE_FLAG = PASS AUTOQA_SURFSLOPE_FLAG = PASS SURFELEVATION_PctOOB = N/A SURFROUGHNESS_PctOOB = N/A SURFREFLECTANCE_PctOOB = N/A SURFSLOPE_PctOOB = N/A
GLA13.QAPFlags	AUTOQA_SURFELEVATION_FLAG = PASS AUTOQA_SURFREFLECTANCE_FLAG = PASS SURFELEVATION_PctOOB = N/A SURFROUGHNESS_PctOOB = N/A

Table 21-2 Sample QAPFLAGS Files

GLA14.QAPFlags	AUTOQA_SURFELEVATION_FLAG = PASS AUTOQA_SURFROUGHNESS_FLAG = PASS AUTOQA_SURFREFLECTANCE_FLAG = PASS AUTOQA_SURFSLOPE_FLAG = PASS SURFELEVATION_PctOOB = N/A SURFROUGHNESS_PctOOB = N/A SURFREFLECTANCE_PctOOB = N/A SURFSLOPE_PctOOB = N/A
GLA15.QAPFlags	AUTOQA_SURFELEVATION_FLAG = PASS AUTOQA_SURFROUGHNESS_FLAG = PASS AUTOQA_SURFREFLECTANCE_FLAG = PASS SURFELEVATION_PctOOB = N/A SURFROUGHNESS_PctOOB = N/A SURFREFLECTANCE_PctOOB = N/A

21.8.2 VAV file

The VAV file has four sections. The sections are separated by lines of dashes. A sample VAV file is given after the descriptions.

21.8.3 VAV file section 1

The file starts with the names of the files being compared, followed by a description of the contents of sections 2 and 3, and ends with the number of matching along-track records.

21.8.4 VAV file Section 2

Section 2 starts with the line "ALONG-TRACK STATISTICS:". This is followed by statistics summarizing the data in each parameter and a comparison of the data in the two files. All along-track parameters are vectors. A sample of the beginning of this section is given below. The four lines of information for each parameter contain

Line 1: The AVERAGE value of the parameter, the Min, the Max, and the Standard Deviation for file 1.

Line 2: The same, for file 2

Line 3: The difference: File 2 - File 1

Line 4: The % difference: 100 * diff / File 2 value

21.8.5 VAV file Section 3

Section 3 starts with the line "SUMMARY STATISTICS:". This is followed by statistics summarizing the data in each parameter in the QAP summary record and a comparison of the data in the two files. Summary records contain both scalar and array data. All array data are treated as vectors. A sample of the beginning of this section is given below.

The information for array data has the same format as for vectors. See section 10.2.2 for a description. For scalar variables, the output consists of one line containing (in order)

- the name of the parameter
- the value from the QAP file
- the value from the QVV file
- the difference (QAP-QVV)
- the % difference (100 (QAP-QVV) / QVV)

21.8.6 VAV file Section 4

Section 4 summarizes the comparison of the two QAP files. It starts with a description of the comparisons. A sample section is given below.

The statistics are:

1) The avg, min, max, and standard deviation of the % differences for all items for which a comparison was made.
2) The number of parameters included in 1.
3) The maximum number of parameters that could have been included in 1. This excludes items not available in output from QAPG.
4) The number of items that can be calculated from the product alone (i.e., the number of items available in QAP files written by QAPG).
5) The number of items with good data for which the % difference could not be calculated because the file 2 value was zero (so would cause a divide by zero error).
6) The number of discrepancies (unexpectedly large differences).
7) The number of parameters that were invalid unexpectedly.
8) The number of along-track records for which the times or locations did not agree.

Table 21-3 Sample VAV File

```
COMPARING CONTENTS OF
   FILE 1: /XOVER/disk4/jack/glas/Mantis795/
GLA01_012_1102_002_0013_4_01_0001.QAP
   FILE 2: /XOVER/disk4/jack/glas/Mantis795/
GLA01_012_1102_002_0013_4_01_qapg_0001.qap
For vector data: Average, Min, Max, Standard deviation of
           Line 1: File 1 data
           Line 2: File 2 data
           Line 3: Diff = File 2 data - File 1 data
           Line 4: 100 * Diff / File 2 data
For scalar data: File 1 datum, File 2 datum, Diff = File 2 datum - File 1 datum,
                 100 * Diff / File 2 datum
Not Avail = the item is not present in one or both files and it can not be
            calculated from the product file alone.
No Data = the item is not present in one or both files. It should be
          possible to calculate it from the product file alone
Invalid = the value is zero in File 2 so the ratio is undefined.
+++ = bad data found for File 1
XXX = bad data found for File 2
*** = bad data found for both files
OOO = problem with comparison of File 1 and File 2 for this item
NUMBER OF MATCHING ALONG-TRACK RECORDS:          101
-----------------------------------------------------------
ALONG-TRACK STATISTICS:
DQATIME1:                     9.94E+07   9.94E+07   9.94E+07   4.69E+02
                              9.94E+07   9.94E+07   9.94E+07   4.69E+02
                             -3.26E-08  -4.77E-07   5.07E-07   2.85E-07
                             -3.28E-14  -4.80E-13   5.10E-13   2.87E-13

...

I_PREDOM_WAVE:                6.34E-01   0.00E+00   1.00E+00   4.84E-01
                              6.34E-01   0.00E+00   1.00E+00   4.84E-01
                              0.00E+00   0.00E+00   0.00E+00   0.00E+00
                              0.00E+00   0.00E+00   0.00E+00   0.00E+00

-----------------------------------------------------------
```

Table 21-3 Sample VAV File

```
SUMMARY STATISTICS:
DQATIME1:                     9.94E+07  Not Avail  Not Avail  Not Avail  XXX
DQATIME2:                     9.94E+07  Not Avail  Not Avail  Not Avail  XXX
I_EXP_APID19:                     1617  Not Avail  Not Avail  Not Avail  XXX
I_NUM_APID19:                     1616       1616          0   0.00
...
I_LFILTNUM_CNT:               4.90E+03   1.40E+03   1.05E+04   3.16E+03
                              4.90E+03   1.40E+03   1.05E+04   3.16E+03
                              0.00E+00   0.00E+00   0.00E+00   0.00E+00
                              0.00E+00   0.00E+00   0.00E+00   0.00E+00

I_SFILTNUM_CNT:               1.27E+03   1.15E+02   4.90E+03   1.81E+03
                              1.27E+03   1.15E+02   4.90E+03   1.81E+03
                              0.00E+00   0.00E+00   0.00E+00   0.00E+00
                              0.00E+00   0.00E+00   0.00E+00   0.00E+00
-----------------------------------------------------------------------
Summary information
Summary:           avg, min, max, std dev of the % differences
N in Summary:      Number of points in summary statistics. This can be different
                   from the "Max N in summary" for several reasons:
                   a. The % diff cannot be calculated if the quantity from file
                      2 is 0 while the quantity from file 1 is not exactly 0, so
                      the ratio is not defined. The number of such items is
                      given seperately as "N Zero."
                   b. Items that cannot be calculated from the product alone
                      will be missing if one of the QAP files was generated
                      after production processing.
                   c. If the values are invalid in one or both QAP files, the
                      item is not included in the total. If a value is invalid
                      in only one file, and should be present in both (see b),
                      this is a discrepency.
Max N in summary:  Maximum number of items that could be in the summary.
N Not in Product:  Number of items that cannot be calculated from data in the
                   product file alone
N Zero:            Number of items with good data for which the % difference
                   could not be calculated because the value for file 2 (but not
                   file 1) was zero, making the ratio infinite.
N Discrepencies:   Number of items for which an unexpected difference was found
N Bad Data Found:  Number of items for which unexpected bad data were found
N Non-matching:    Number of along-track records for which the start/end
                   times (QAP1,2,5) or lat/lon (QAP6,12-15) are different.
                   Not applicable to other products.
```

Table 21-3 Sample VAV File

```
SUMMARY:                      -4.68E-03  -3.81E+01   2.30E+01   4.69E-02
MAX N IN SUMMARY:            84
N NOT IN PRODUCT:            10
N ZERO:                       0
N DISCREPENCIES:              0
N BAD DATA FOUND:            11
N IN SUMMARY:                73
N NON-MATCHING:               0

11.0 Error and Warning Messages
```

21.9 Error and Warning Messages

All warning and error messages generated by the program are self-explanatory. Any error messages generated by IDL should be reported.

In addition to errors and warnings written to stderr and stdout, the program generates an exit code. The exit codes used are specified in /vobs/glas/idl/qa_browse/util/qa_exitcodes.pro.

21.10 Recovery Steps

Exit IDL, correct the error in the control file, and start over.

Section 22

Creating a Global DEM and Replacing Greenland and Antarctica DEM with New Grids

1) Download the SRTM30 data from http://edcftp.cr.usgs.gov/pub/data/srtm/SRTM30/

e020n40.dem.zip

e060n40.dem.zip

e100n40.dem.zip

e140n40.dem.zip

w020n40.dem.zip

w060n40.dem.zip

w100n40.dem.zip

w140n40.dem.zip

w180n40.dem.zip

e020n90.dem.zip

e060n90.dem.zip

e100n90.dem.zip

e140n90.dem.zip

w020n90.dem.zip

w060n90.dem.zip

w100n90.dem.zip

w140n90.dem.zip

w180n90.dem.zip

e020s10.dem.zip

e060s10.dem.zip

e100s10.dem.zip

e140s10.dem.zip

w020s10.dem.zip

w060s10.dem.zip

w100s10.dem.zip

w140s10.dem.zip

w180s10.dem.zip

2) Download the Antarctica data from http://edcdaac.usgs.gov/gtopo30/gtopo30.html

   w060s60.tar.gz

   w180s60.tar.gz

   w120s60.tar.gz

   w000s60.tar.gz

   e060s60.tar.gz

   e120s60.tar.gz

3) Ftp all the files to the working directory

   All in the working directory

4) Unzip the zip files

   (unzip for *.zip,   gunzip and tar xvf for *.tar.gz)

5) Run the following programs:

   merge40 to merge the data file of latitude –10 to 40 degrees

   merge90 to merge the data file of latitude 40 to 90 degrees

   merges10 to merge the data file of latitude –60 to –10 degrees

   mergesAntar to merge the data file of latitude –90 to –60

6) Run mergeall to merge all the data files to the final file:GLOBAL.dem

7) Run interp_ourgrid_to30secDEM to replace a new grid in Antarctica and Greenland.

   Replacing Antarctica grid:

   interp_ourgrid_to30secDEM /de1/newgrids/ac_gmJGM3_ers1DGM_5km_slcr.GRID-egm96 GLOBAL.dem tempfile

   Replacing Greenland grid:

   interp_ourgrid_to30secDEM /de1/newgrids/gc_gmJGM3_ers1DGM_5km_slcr.GRID-egm96

   tempfile NEW_GLOBAL.dem.

8) Run create_newgrid to add data above 89.995833 and below -89.995833 deg lat

The final file is NEWGLOBAL.dem.

9) Run convrt1deg_new to create the one degree mask identifying region types in them (ocean or land or mix)

   The final file is ONEDEG.MASK.

10) Run cr_1degDem to create DEM.MASK file that has 0 for ocean, and if not 0, i.e. land, then the value is the record number of the new global dem containing that 1 deg segment.

    It also creates LAND.DEM that contains only land dem.

11) Run unit test to read the LAND.DEM and DEM.MASK and compares the data values to a test data in test_dempoints.out.

Section 23

prod_util

23.1 Function

The content of prod_util is a set of non-production product-related utilities. There are five utilities present: scantime, product_test, gsas_prod_readers, strip1984 and stripper.

Only stripper and scantime are based on the standard GSAS PGE model. Therefore, no multi-granule or multi-file support is provided, no time selection is available, and no control files are necessary for the other utilities.

Note: The prod_util routines are not intended for production use.

23.2 scantime

scantime is a utility which reads ANC29, ANC23, APID19 and APID1984 and detects secondary header time resets. This can be used to assist in the generation of ANC33 updates. scantime uses a standard GSAS control file.

Required input files are listed in Table 23-1.

Table 23-1 scantime Inputs

File Spec	Type	Source	Short Description
gla00*_0019.dat	Level-0 APID	EDOS	GLAS Level-0 APID19.
gla00*_1984.dat	Level-0 APID	EDOS	GLAS Level-0 APID1984.
anc07*_0000.dat	Static Ancillary	Science Team	GLAS error file.
anc07*_0001.dat	Static Ancillary	Science Team	GLAS global constants file.
anc25*.dat	Dynamic Ancillary	Science Team	GPS/UTC conversion file.
anc33*.dat	Dynamic Ancillary	ISIPS Operations	Counter-to-UTC conversion file.
Control File	Control	ISIPS Operations	Control file.

Output files are listed in Table 23-2.

Table 23-2 scantime Outputs

File Spec	Type	Destination	Short Description
anc33*.dat	Dynamic Ancillary	ISIPS Operations	Suggested replacement ANC33 file.
anc06*.dat	Dynamic Ancillary	ISIPS Operations	Standard metadata/processing log file.

The control file section delimiter for scantime is:

```
=SCANTIME
```

23.3 product_test

product_test is used to verify product formats in a testing environment. It contains programs which write product data records with minimum, maximum, and intermediate values. Other programs read what has been written to verify that no change was detected in the values read versus the values written. Two executables (glaxx_writer and glaxx_reader) are provided for each GLA filetype.

glaxx_writer requests the name of an output GLAxx product file from the user and writes data records to that file, as well as creating the text-based output files described in Table 23-3.

Table 23-3 glaxx_writer outputs

Output File	Description
filename.write.scal	contains text-format scaling values for the appropriate product.
filename.write.prod	contains text-format product (scaled) values for each data record written.
filename.write.alg	contains text-format algorithm (unscaled) values for each data record written.

glaxx_reader requests the name of an input GLAxx product file from the user and then reads that file to create text-based output files described in Table 23-4.

Table 23-4 glaxx_reader outputs

Output File	Description
filename.read.scal	contains text-format scaling values for the appropriate product.
filename.read.prod	contains text-format product (scaled) values for each data record written.
filename.read.alg	contains text-format algorithm (unscaled) values for each data record written.

Once glaxx_writer creates the output file and glaxx_reader reads it, the resulting text output files can be compared to determine if any data were lost or changed during the input/output process.

23.4 gsas_prod_readers

gsas_prod_readers are sample product readers which are can be distributed independent from the rest of GSAS. These are designed to provide external data users with sample code for reading the GLAS products. The gsas_prod_readers are fully documented in a separate document which can be provided to the data users along with the code.

23.5 strip1984

strip1984 is a small utility which strips extraneous data from the APID1984 packets. This was created because extra information was added to the 1984 packets during GSFC testing. strip1984 requests the name of an input 1984 APID file. It reads the secondary header of the first data record to determine the correct record length and then writes the trimmed records to an output file. The output filename is the input filename with '.stripped' appended to it.

23.6 stripper

stripper is a utility written to help process spacecraft test data. It subsets APIDs by selecting data which fall within the specified MET count range. It is modeled after a full-fledged PGE using a slightly modified control file format (MET counts are substituted for UTC seconds in the time specification).

Appendix A
GLAS File Summary

This Appendix is a summary of the GLAS File types and provides information about the recommended production naming conventions. Full details regarding the GLAS product and ancillary files are provided in the GLAS Science Data Management Plan and the GLAS Standard Data Products Specifications - Level 1 and Level 2.

A.1 GLAS Production File Naming Convention

GSAS creates products and uses various ancillary files and other input products that are track/time dependent, or process-specific and created automatically. Some ancillary files are static and others are dynamic. The following scheme outlines the file naming conventions required for each file type.

1) Track dependent products processed at ISIPS (GLA01-15; and corresponding png files, qap products, v and v files, and metadata files)

   GLAxx_mmm_prkk_ccc_tttt_s_nn_ffff.eee

   Where eee is dat for GLA01-15 and eee is hdf for GLA16; qap for corresponding qap file; png for corresponding non-hdf browse files; vav for corresponding validation and verification file; met for metadata files

   Note that the multiple file pngs for GLA01, 02, 03 and 05-15 will use the ffff to denote the different files

2) Time dependent GLA and ANC granules: (GLA00, dynamic ANC time dependent files except for ANC06)

   GLA00_mmm yyyymmdd_hhmmss_nn_ffff.eee and
   ANCxx_mmm yyyymmdd_hhmmss_nn_ffff.eee

   Where eee is always dat, xx is the corresponding ANC file number

   yyyymmdd_hhmmss is the date_time of the first data point in the file

   for GLA00 the ffff has to have a one to one correspondence with the APID number. It the apid number is less than 4 digits then the unused portion of the ffff filed will be set to 0. The numbers will be filled in from the right most 'f' field, i.e. apid 26 will be denoted as 0026.

3) Static Ancillary files required for processing (ANC)

   ANCxx_mmm_nn_ffff.eee

   Where eee is always dat, xx is the corresponding ANC file number

4) Log files, ANC06

   ANC06_mmm_yyyymmdd_ gggggg _nn_pgename.txt

yyyymmdd is the creation date

5) Control files

CTL_mmm_yyyymmdd_gggggg_nn_pgename.ctl

Where yyyymmdd is the creation date

6) SCF QA files

SCFQA_mmm_yyyymmdd_iiiiii_nn.txt

Where yyyymmdd is the creation date

Time for process-specific files is processing date. GLAS file naming keys are defined in Table A-1.

Table A-1 GLAS File Naming Keys

Key	Description
xx	Type ID number (CCB assigned number within a specific GLA or ANC series)
p	repeat ground track phase
r	reference orbit number
kk	instance # incremented every time GLAS enters a different reference orbit
ccc	cycle of reference orbit for this phase
tttt	track within reference orbit
s	segment of orbit. This is 0 on files that contain multiple segments (GLA02, GLA03, GLA04,GLA07-GLA15) and 1,2,3,or 4 on GLA01, GLA05, and GLA06.
yyyymmdd	starting date in year, month, and day of month or creation date (see above)
hhmmss	starting time hour, minute, second
mmm	release number for process that created the produce (CCB assigned-combination of software and data)
nn	granule version number (the number of times this granule is created for a specific release)
iii	counting sequence number (incremental sequence per day for each instance of a process specific ANCxx or GLAxx)
ffff	file type (numerical,CCB assigned for multiple files as needed for data of same time period for a specific ANCxx or GLAxx, .i.e. multi-file granule)
eee	descriptor telling whether data product, browse product, quality assurance product, validation and verification output. Valid eee values are dat, png, hdf, qap, vav, and met
dddd	'brws' denotes consolidated multi-png file browse product in hdf

Table A-1 GLAS File Naming Keys (Continued)

Key	Description
gggggg	job id number – number of digits may change.
pgename	GLAS_ATM, GLAS_L0P, met_util, GLAS_L1A, GLAS_Meta, ATM_ANC, met_util, GLAS_ALT, QABROWSE, ExtractRev

A.2 EDOS L0 File Naming Convention

EDOS-delivered files have a different naming convention than standard GLAS product files. The EDOS L0 files will be renamed as part of the ingest process. Incoming filenames are as follows:

```
fooooooottttttthhhhhhhdddddddddddiuu.eeee
```

Table A-2 defines keys for the EDOS file naming convention.

Table A-2 EDOS L0 File Naming Keys

Key	Description
f	E (for EDS) or P (for PDS) (1byte)
ooooooo	First APID in data set (SCID 3 bytes, APID 4 bytes)
ttttttt	Second APID in data set (SCID 3 bytes, APID 4 bytes)
hhhhhhh	Third APID in data set (SCID 3 bytes, APID 4 bytes)
ddddddddddd	creation GMT/ZULU time (11 bytes)
I	numeric identification (0-9) (1 byte)
uu	Unique file number (00-99) (2 bytes)
eeee	File naming extension PDS or EDS (4 bytes)

A.3 GLAS File Types

Table A-3 lists GLAS file IDs, a short description, and file type for each file used in GSAS. Note that QAP files, which correspond to each product file, are not listed. The "multi" column indicates if a file type is a multi-file granule. These means there are multiple subtypes within each file type.

Table A-3 GLAS File Types

File Type	Short Description	Type	Multi
ANC01	Meteorological Data	Dynamic Anc	Yes 00=header 01=precip water 02=height 03=rel humidity 04=temperature
ANC03	Laser Tracking Data	Dynamic Anc	No
ANC04	IERS Rotation	Dynamic Anc	No
ANC05	Magnetic and Solar Flux	Dynamic Anc	No
ANC06	GLAS Metadata and Log	Dynamic Anc	No
ANC07	GLAS Coefficients and Constants	Static Anc	Yes 00=errors 01=global 02=atmosphere 03=elevation 04=waveform 05=L1A 06=utility
ANC08	Precision Orbit Data	Dynamic Anc	No
ANC09	Precision Attitude Data	Dynamic Anc	No
ANC10	GPS Tracking	Dynamic Anc	No
ANC11	Instrument Trend	Dynamic Anc	No
ANC12	Digital Elevation Model	Static Anc	Yes 00=DEM 01=Mask
ANC13	Geoid	Static Anc	No
ANC16	Load Tide Model	Static Anc	No
ANC17	Ocean Tide Model	Static Anc	No
ANC18	Standard Atmosphere	Static Anc	No
ANC20	Predicted Orbit	Dynamic Anc	No
ANC21	Science Trend Data	Dynamic Anc	No
ANC22	Reference orbit	Dynamic Anc	No
ANC23	Reference ascending node	Dynamic Anc	No
ANC25	GPS-UTC time	Dynamic Anc	No

Table A-3 GLAS File Types (Continued)

File Type	Short Description	Type	Multi
ANC27	Regional masks	Static Anc	No
ANC28	Rev node table	Dynamic Anc	No
ANC29	Index	Dynamic Anc	No
ANC30	Global aerosol categorization map	Static Anc	No
ANC31	Aerosol tropospheric classification map)	Static Anc	No
ANC32	GPS time correction	Dynamic Anc	No
ANC33	MET Counter to UTC conversion	Dynamic Anc	No
ANC34	BCTCW counter to UTC conversion	Dynamic Anc	No
ANC35	Ozone	Static Anc	No
ANC36	Atmosphere Utility	Dynamic Anc	No
ANC37	Spacecraft CG	Dynamic Anc	No
ANC38	Multiple-scattering table	Dynamic Anc	No
ANC39	GPS Data	Dynamic Anc	No
ANC40	Raw Meteorological File	Dynamic Anc	No
ANC41	JPL Planetary Ephemeris	Dynamic Anc	No
ANC42	Reference Orbit Table	Dynamic Anc	No
ANC43	Track File	Dynamic Anc	No
ANC44	Atm 1064 Cirrus CAL File	Dynamic Anc	No
ANC45	Product Metadata template files	Static Anc	Yes 01=GLA01 02=GLA02 etc...
ANC46	Ancillary Metadata template files	Static Anc	Yes 04=ANC04 08=ANC08 etc...
ANC47	EDOS Construction Records	Dynamic ANC	yes (one file per APID type)
GLA00	GLAS Instrument Packet File (APIDs)	Product	yes (one file per APID type)
GLA01	Altimetry Data	Product	No
GLA02	Atmosphere Data	Product	No
GLA03	Engineering Data	Product	No

Table A-3 GLAS File Types (Continued)

File Type	Short Description	Type	Multi
GLA04	SRS and GPS Data	Product	Yes 01=LPA 02=LRS 03=GYRO 04=IST 05=BST 06=SCPA
GLA05	Waveform-based Range Corrections	Product	No
GLA06	Elevation	Product	No
GLA07	Backscatter	Product	No
GLA08	Boundary Layer and Aerosol Layer Heights	Product	No
GLA09	Cloud Height for Multiple Layers	Product	No
GLA10	Aerosol Vertical Structure	Product	No
GLA11	Thin Cloud/Aerosol Optical Depth	Product	No
GLA12	Ice Sheet Products	Product	No
GLA13	Sea Ice Products	Product	No
GLA14	Land Products	Product	No
GLA15	Ocean Products	Product	No

A.4 GLAS File Associations

Table A-4 lists each GLAS file and its associated source or PGE. Those inputs in *italics* are for special processing scenarios. Note that GLAS_Reader, though not listed as a destination, can read most of these files.

Table A-4 GSAS File Associations

File Type	Short Description	Source (output from)	Destination (input to)
ANC01	Meteorological Data	met_util	GLAS_Alt GLAS_Atm
ANC04	IERS Polar Motion and Earth Rotation Data File.	UTexas	GLAS_Alt, GLAS_Atm
ANC06	GLAS Metadata and Log	All core PGEs	Archive
ANC07	GLAS Coefficients and Constants	Science Team	All PGEs
ANC08	Precision Orbit Data	UTexas	GLAS_Alt GLAS_Atm

Table A-4 GSAS File Associations (Continued)

File Type	Short Description	Source (output from)	Destination (input to)
ANC09	Precision Attitude Data	UTexas	GLAS_Alt GLAS_Atm
ANC12	Digital Elevation Model	Science Team	GLAS_Alt GLAS_Atm
ANC13	Geoid	Science Team	GLAS_Alt GLAS_Atm
ANC16	Load Tide Model	Science Team	GLAS_Alt
ANC17	Ocean Tide Model	Science Team	GLAS_Alt
ANC18	Standard Atmosphere	Science Team	GLAS_Atm
ANC20	Predicted Orbit	UTexas	createGran, GLAS_L1A, GLAS_Alt, GLAS_Atm
ANC25	GPS-UTC time	UTexas	GLAS_L1A
ANC27	Regional masks	Science Team	GLAS_Atm
ANC29	Index files	GLAS_L0proc	GLAS_L1A
ANC30	Global aerosol categorization map	Science Team	GLAS_Atm
ANC31	Aerosol tropospheric classification map)	Science Team	GLAS_Atm
ANC32	GPS time correction	GLAS_L0proc	GLAS_L1A
ANC33	MET Counter to UTC conversion	Instrument Team	GLAS_L0proc
ANC35	Ozone file	Science Team	Atm_anc
ANC36	Atmosphere Utility	Atm_anc	GLAS_Atm
ANC40	Raw meteorological file.	GSFC DAAC	met_util
ANC44	Atm 1064 Cirrus CAL File	atm_anc	Science Team
ANC45	Product Metadata template files	Science Team	GLAS_L1A GLAS_Atm GLAS_Alt GLAS_Meta
ANC46	Ancillary Metadata template files	Science Team	GLAS_Meta
ANC47	EDOS Construction Records	EDOS	GLAS_L0proc
GLA00	GLAS Instrument Packet File (APIDs)	EDOS	GLAS_L0proc, GLAS_L1A
GLA01	Altimetry Data	GLAS_L1A	GLAS_Alt, Archive

Table A-4 GSAS File Associations (Continued)

File Type	Short Description	Source (output from)	Destination (input to)
GLA02	Atmosphere Data	GLAS_L1A	GLAS_Atm, atm_anc, Archive
GLA03	Engineering Data	GLAS_L1A	GLAS_Tick, Archive
GLA04	SRS and GPS Data	GLAS_L1A	GLAS_L1A, UTexas, Archive
GLA05	Waveform-based Range Corrections	GLAS_Alt	GLAS_Alt, Archive
GLA06	Elevation	GLAS_Alt	GLAS_Alt, Archive
GLA07	Backscatter	GLAS_Atm	GLAS_Atm, Archive
GLA08	Boundary Layer and Elevated Aerosol Layer Heights	GLAS_Atm	Archive
GLA09	Cloud Height for Multiple Layers	GLAS_Atm	GLAS_Alt, Archive'
GLA10	Aerosol Vertical Structure	GLAS_Atm	Archive
GLA11	Thin Cloud/Aerosol Optical Depth	GLAS_Atm	GLAS_Alt, Archive
GLA12	Ice Sheet Products	GLAS_Alt	Archive'
GLA13	Sea Ice Products	GLAS_Alt	Archive
GLA14	Land Products	GLAS_Alt	Archive'
GLA15	Ocean Products	GLAS_Alt	Archive

Appendix B

ANC07 Format, Files and Contents

B.1 ANC07 Format

ANC07 files are in standard GSAS "keyword=value" format. ANC07 files contain sections for errors messages, status messages, global constants, and subsystem-specific constants. Sections may be combined into a single file or separated into multiple files. Not all sections need be present. The sections are delimited as follows:

```
BEG_OF_STATUS=
...Status section contents...
END_OF_STATUS=

BEG_OF_ERROR=
...Error section contents...
END_OF_ERROR

BEG_OF_GLOBALS=
...Global constants section contents...
END_OF_GLOBALS

BEG_OF_ATM=
...Atmosphere constants section contents...
END_OF_ATM

BEG_OF_ELEV=
...Elevation constants section contents...
END_OF_ELEV

BEG_OF_L1A=
...L1A constants section contents...
END_OF_L1A

BEG_OF_UTIL=
...Utility constants section contents...
END_OF_UTIL
```

B.2 ANC07 Files

In production, ANC07 is defined as a multi-granule file with a single granule representing a different section (for the most part). Table B-1 lists the granule names and contents. See Appendix A for more information regarding file naming conventions.

Table B-1 ANC07 Granule Names and Contents

Naming Convention	Contents
anc07_mmm_nn_0000.dat	Error and Status sections
anc07_mmm_nn_0001.dat	Global constants
anc07_mmm_nn_0002.dat	Atmosphere constants
anc07_mmm_nn_0003.dat	Elevation constants
anc07_mmm_nn_0004.dat	Waveform constants
anc07_mmm_nn_0005.dat	L1A constants
anc07_mmm_nn_0006.dat	Utility constants

B.3 ANC07 Contents

Global constants and those used by the L1A, Atmosphere, Elevation, and Waveforms are listed in this section. Error and Status messages are listed in Appendix C. Values listed here are current as of V2.2

B.3.1 Global Constants

```
GLOB_VERS = ANC07 Globals V3.0 October 2002
#
# Mean Ellipsoid Radius
#
ELLIPAE = 6378136.3d0
#
# Ellipsiod Flattening
#
ELLIPF = 298.257d0
#
# delta lat for DEM grid
#
DEMdLat  =  0.00833333333333d0
#
# delta lon for DEM grid
#
DEMdLon =  0.00833333333333d0
#
# Start latitude on the DEM grid
#
DEMLAT_BEG  = 90.00416666666667d0
#
# Start longitude on the DEM grid
#
DEMLON_BEG =  -0.00416666666667d0
```

```
#
# The offset from the GPS reference time (00:00 Jan 1, 1980)
# to the UTC reference time (12:00 Jan 1, 2000)
#
#GPStoUTCoffset =  -1001.10980499199991
#GPStoUTCoffset = -46197.10980d0
#
# (correct)
#
GPStoUTCoffset = -630763200.00000
#
# For v2.2 we are adding a 2.75 second bias to the offset to make recndx
# align with utctime. This will be investigated further any may actually
# be a modification to the offset table used with MET alignment.
#
#GPStoUTCoffset = -630763202.75000
#
# Max PAD offset allowed (max off-nadir pointing angle)
#
MAXPADOFF =  0.5d0
#
# 1064 energy Calibration coefficient for Received WFs
#
gd_calibcoefrec = 5.88d-20
#
# 1064 energy Calibration coefficient for transmitted  WFs
#
gd_calibcoeftr = 3.38d-5
```

B.3.2 L1A Constants

```
L1A_VERS = ANC07 L1a V3.0 October 2002
#
# 1064 transmit energy default to use when computed energy is unavailable
#
GD_IR_TXNRG = 70.0D-3
#
# 532 / 1064 Lower energy threshold (Joules)
#
GD_NRG_THR1 = 1.0D-3
#
# 532 / 1064 Upper energy threshold (Joules)
#
GD_NRG_THR2 = 100.0D-3
#
# 532 energy mJ per count
#
GD_G_NRG_MJ_CT = 0.25d0
#
# Default value for range from satellite to 532 profile (m)
#
GD_RNG2PCPROF = 5.6d5
#
# Lower threshold for range from satellite to 532 profile (m)
#
GD_RNG2PCPROF1 = 5.0d5
#
# Upper threshold for range from satellite to 532 profile (m)
```

```
#
GD_RNG2PCPROF2 = 6.2d5
#
# ground ret threshold
#
GI_GRND_THRESH = 20
#
#  cloud ret threshold
#
GI_CLD_THRESH = 5
#
#lidar bin sat val
#
GI_LID_SATVAL = 80
#
# quality flag(1hz) for integrated return
#
GI_INT_QFVAL= 800,600,400,100
#
# 532 laser energy max value
#
GI_G_NRGMAX = 36
#
# 1064 laser energy max value
#
GI_IR_NRGMAX = 73
#
#background index used for 532/1064
#1=upper, 2=lower, 3=(upper+lower)/2, 4=avg of last 8 bins of the profile
#
GI_G_USEDBG = 1
#
# background index used for 532/1064
#
GI_IR_USEDBG = 3
#
# GLA04 time latencies
#
GD_GYRO_TIME_LAT = 0.0D0
GD_BST1_TIME_LAT = 0.0D0
GD_BST2_TIME_LAT = 0.0D0
GD_LRS_TIME_LAT = 0.03274d0
GD_IST_TIME_LAT = 0.032168d0
GD_POS_ATT_TIME_LAT = 0.0D0
#
# deadtime correction lookup table  - allows for up to 255 entries in the table
#    note : each keyword value can not exceed 255 characters therefore value
#    distributed between several keywords.
#
D_DTC_TBL1=9.9505D-01, 1.0000D+00, 1.0018D+00, 1.0091D+00, 1.0154D+00, 1.0206D+00,
1.0286D+00, 1.0373D+00, 1.0457D+00, 1.0531D+00, 1.0602D+00, 1.0673D+00, 1.0745D+00,
1.0818D+00, 1.0895D+00, 1.0975D+00, 1.1055D+00, 1.1137D+00, 1.1224D+00, 1.1310D+00
D_DTC_TBL2=1.1396D+00, 1.1486D+00, 1.1575D+00, 1.1664D+00, 1.1756D+00, 1.1850D+00,
1.1947D+00, 1.2044D+00, 1.2142D+00, 1.2242D+00, 1.2343D+00, 1.2446D+00, 1.2550D+00,
1.2654D+00, 1.2759D+00, 1.2865D+00, 1.2971D+00, 1.3085D+00, 1.3202D+00, 1.3326D+00
D_DTC_TBL3=1.3452D+00, 1.3578D+00, 1.3705D+00, 1.3832D+00, 1.3960D+00, 1.4092D+00,
1.4226D+00, 1.4361D+00, 1.4497D+00, 1.4634D+00, 1.4777D+00, 1.4928D+00, 1.5080D+00,
1.5236D+00, 1.5393D+00, 1.5553D+00, 1.5714D+00, 1.5876D+00, 1.6041D+00, 1.6206D+00
D_DTC_TBL4=1.6371D+00, 1.6536D+00, 1.6702D+00, 1.6880D+00, 1.7066D+00, 1.7257D+00,
```

```
1.7452D+00, 1.7648D+00, 1.7845D+00, 1.8046D+00, 1.8295D+00, 1.8549D+00, 1.8807D+00,
1.9068D+00, 1.9333D+00, 1.9637D+00, 1.9961D+00, 2.0287D+00, 2.0612D+00, 2.0939D+00
D_DTC_TBL5=2.1290D+00, 2.1656D+00, 2.2028D+00, 2.2408D+00, 2.2795D+00, 2.3230D+00,
2.3681D+00, 2.4134D+00, 2.4590D+00, 2.5052D+00, 2.5566D+00, 2.6086D+00, 2.6616D+00,
2.7146D+00, 2.7738D+00, 2.8408D+00, 2.9085D+00, 2.9774D+00, 3.0463D+00, 3.1191D+00
D_DTC_TBL6=3.2012D+00, 3.2837D+00, 3.3673D+00, 3.4508D+00, 3.4930D+00, 3.5231D+00,
3.5529D+00, 3.5837D+00, 3.6157D+00, 3.6477D+00, 3.6799D+00, 3.7132D+00, 3.7469D+00,
3.7818D+00, 3.8163D+00, 3.8427D+00, 3.8690D+00, 3.8953D+00, 3.9219D+00, 3.9453D+00
D_DTC_TBL7=3.9702D+00, 3.9963D+00, 4.0234D+00, 4.0531D+00, 4.0829D+00, 4.1126D+00,
4.1423D+00, 4.1721D+00, 4.2023D+00, 4.2365D+00, 4.2712D+00, 4.3071D+00, 4.3459D+00,
4.3858D+00, 4.4265D+00, 4.4674D+00, 4.5096D+00, 4.5567D+00, 4.6039D+00, 4.6538D+00
D_DTC_TBL8=4.7079D+00, 4.7622D+00, 4.8165D+00, 4.8786D+00, 4.9466D+00, 5.0147D+00,
5.0829D+00, 5.1566D+00, 5.2438D+00, 5.3448D+00, 5.4459D+00, 5.5470D+00, 5.6527D+00,
5.7655D+00, 5.8916D+00, 6.0178D+00, 6.1602D+00, 6.3285D+00, 6.4970D+00, 6.6802D+00
D_DTC_TBL9=6.9390D+00, 7.1982D+00, 7.5347D+00, 7.9302D+00, 8.6148D+00, 8.9366D+00,
9.0927D+00, 9.2594D+00, 9.5340D+00, 9.8213D+00, 1.0149D+01, 1.0611D+01, 1.1255D+01,
1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01
D_DTC_TBL10=1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01,
1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01,
1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01, 1.1259D+01
D_DTC_TBL11=0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0,
0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0,
0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0
D_DTC_TBL12=0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0,
0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0,
0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0
D_DTC_TBL13=0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0,
0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0, 0.0000D0
#
# Maximum LIDAR value for 532 nm
#
gD_G_MAXLID = 200.0d0
#
# digital to analog convrsn lookup table
#
D_D2A_TBL1=0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
D_D2A_TBL2=21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39,
40, 41
D_D2A_TBL3=42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60,
61, 62
D_D2A_TBL4=63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81,
82, 83
D_D2A_TBL5=84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101,
102, 103
D_D2A_TBL6=104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118,
119, 120
D_D2A_TBL7=121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135,
136, 137
D_D2A_TBL8=138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152,
153, 154
D_D2A_TBL9=155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169,
170, 171
D_D2A_TBL10=172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186,
187, 188
D_D2A_TBL11=189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203,
204, 205
D_D2A_TBL12=206, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219,
220, 221
D_D2A_TBL13=222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236,
```

```
237, 238
D_D2A_TBL14=239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253,
254,255
#
# Maximum LIDAR value for 1064 nm
#
GD_IR_MAXLID = 255.0d0
#
# L_Eng constants
#
GD_AP21_CONST1 = 9.22D0
GD_AP21_CONST2 = 10.0D0
GD_AP21_CONST3 = 1.52D0
GD_AP21_CONST4 = 4.0D0
GD_AP21_CONST5 = 0.408D0
GD_AP21_CONST6 = 2.0D0
GD_AP21_CONST7 = 0.41D0
GD_AP21_CONST8 = 3.0D0
GD_AP21_CONST9 = 0.407D0
GD_AP21_CONST10 = 9.2D0
GD_AP21_CONST11 = 8.0D0
GD_AP21_CONST12 = 1.25D0
GD_AP21_CONST13 = 9.25D0
GD_AP21_CONST14 = 13.0D0
GD_AP21_CONST15 = 1.514D0
GD_AP21_CONST16 = 1.91D0
GD_AP21_CONST17 = 3.52D0
GD_AP21_CONST18 = 0.66D0
GD_AP21_CONST19 = -3.515D0
GD_AP21_CONST20 = 0.63D0
GD_AP21_CONST21 = 1.515D0
GD_AP21_CONST22 = -1.532D0
GD_AP21_CONST23 = 1.49D0
GD_AP21_CONST24 = -1.52D0
GD_AP21_CONST25 = 2.05D0
GD_AP21_CONST26 = 4.05D0
GD_AP21_CONST27 = -4.028D0
GD_AP21_CONST28 = 0.054D0
GD_AP21_CONST29 = 7.0D0
GD_AP21_CONST30 = 0.052D0
GD_AP21_CONST31 = 5.0D0
GD_AP21_CONST32 = 30.0D0
GD_AP21_CONST33 = 30.2D0
#
# L_ENG COEF ARRAYS
#
GD_LSROSC_COEF = 0.5389D0,0.4878D0
GD_LSRAMP_COEF = 0.703D0,0.4865D0
GD_LSRDR_COEF = 131.08D0,0.512D0
GD_ADDETGN_COEF = -1D0,0.0078125D0
GD_DPINA_COEF = 0.0065D0, 0.3823,3.848D-5
GD_DPINB_COEF = -0.0119D0,0.4143,-8.671D-5
GD_532NRG_COEF = -0.122D0,0.4095,-6.601D-05
GD_SPCM_COEF = 0.0D0,3.581D0
GD_INTTMP1_COEF = -50.0D0,0.781D0
GD_INTTMP2_COEF = 9.0D0,0.031D0
GD_VC_COEF = -100.0D0,0.048828125D0
GD_XY_COEF = -10.0D0,0.0048828125D0
GD_AP21_1_COEF = 0.0D0,0.026D0
```

```
GD_AP21_2_COEF = 0.0D0,0.052D0
GD_AP22_1_COEF = -20.4D0,0.3984D0
GD_AP22_2_COEF = -23.5D0,0.3984D0
GD_AP22_3_COEF = -21.6D0,0.3984D0
GD_AP22_4_COEF = -21.0D0,0.3984D0
GD_AP22_5_COEF = -20.7D0,0.3984D0
GD_AP22_6_COEF = -18.113D0,0.3083D0
GD_AP22_7_COEF = 0D0,0.2949D0
GD_AP23_1_COEF = -909.090909D0,0.443892045D0
GD_AP23_2_COEF = -2.0D0,976.5625D-6
GD_AP23_3_COEF = 0.16D0,0.1027D0,-4.253D-05,3.833D-07
GD_AP23_4_COEF = 0.03D0,0.1051D0,-6.469D-05,4.376D-07
GD_AP23_5_COEF = 0.14D0,0.104D0,-5.962D-05,4.304D-07
GD_AP23_6_COEF = 29.27D0,0.09251D0,9.19D-06,1.022D-07
GD_AP23_7_COEF = 0.03D0,0.1173D0,-6.871D-05,2.629D-07
GD_AP23_8_COEF = -7.7D0,0.11D0,-5.1D-05,2.007D-07
GD_AP50_1_COEF = 40D0,-0.15625D0
GD_AP50_2_COEF = 0.16D0,0.1027D0,-4.253D-05,3.833D-07
GD_AP50_3_COEF = 0.14D0,0.104D0,-5.962D-05,4.304D-07
GD_AP50_4_COEF = 0.03D0,0.1051D0,-6.468D-05,4.376D-07
GD_AP50_5_COEF = 29.27D0,0.09251D0,9.19D-06,1.022D-07
GD_AP50_6_COEF = 0.03D0,0.1173D0,-6.871D-05,2.629D-07
GD_AP50_7_COEF = -7.7D0,0.11D0,-5.1D-05,2.007D-07
GD_AP50_8_COEF = 0.0008D0,0.00368D0
GD_AP50_9_COEF = -0.0008D0,0.00311D0
GD_AP50_10_COEF = 0.0008D0,0.00311D0
GD_AP50_11_COEF = -0.0002D0,0.00347D0
GD_AP50_12_COEF = -33.84D0,0.5368D0,-1.622D-3,3.155D-6
GD_AP50_13_COEF = -2.406D0,0.06459D0,-7.58D-6,5.591D-8
GD_AP50_14_COEF = 13.33D0,0.06518D0,-5.261D-6,4.076D-8
#
# Pseudo Telemetry eq. constants
#
PSETLM_F0_N = -273.15D0
PSETLM_F1_N = 1.0185D-3
PSETLM_F2_N = 2.4090D-4
PSETLM_F3_N = 1.4990D-7
PSETTEMP1_N = 5.0D0
PSETTEMP2_N = 7.5D0
PSETTEMP1_D = 255.0D0
PSETTEMP2_D = 7680.0D0
PSETTEMP1 = 0.0D0
PSETTEMP2 = 0.0D0
#
# 28V Bus A Instrument (Volts)
#
GD_BA28INST_SF = 0.0d0, 9.22d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# Hybrid  Supplies (Amps)
#
GD_HYBSUPP_SF = 0.0d0, 1.9d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# HVPS Detector Supplies(Amps)
#
GD_HVPSDET_SF = 0.0d0, 0.405d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# Operational Heaters (Amps)
#
GD_OPHTRS_SF = 0.0d0, 0.395d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
```

```
#
# Mechanical System (Amps)
#
GD_MECHSYS_SF = 0.0d0, 0.395d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 28V Bus B Laser 1 (Volts)
#
GD_BB28L1V_SF = 0.0d0, 9.35d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 28V Bus B Laser 1 (Amps)
#
GD_BB28L1A_SF = 0.0d0, 1.551d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 28V Bus C Laser 2 (Volts)
#
GD_BC28L2V_SF = 0.0d0, 9.35d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 28V Bus C Laser 2 (Amps)
#
GD_BC28L2A_SF = 0.0d0, 1.511d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 28V Bus D Laser 3 (Volts)
#
GD_BD28L3V_SF = 0.0d0, 9.35d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 28V Bus D Laser 3 (Amps)
#
GD_BD28L3A_SF = 0.0d0, 1.555d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 5V Hybrid #1 (Volts)
#
GD_5VHYB1V_SF = 0.0d0, 1.54d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
#  5V Hybrid #1 (Amps)
#
GD_5VHYB1A_SF = 0.0d0, 1.91d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 12V Hybrid #2 (Volts)
#
GD_12VHYB2V_SF = 0.0d0, 3.52d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 12V Hybrid #2 (Amps)
#
GD_12VHYB2A_SF = 0.0d0, 0.66d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# m12V Hybrid #3 (Volts)
#
GD_M12VHYB3V_SF = 0.0d0, -3.52d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# m12V Hybrid #3 (Amps)
#
GD_M12VHYB3A_SF = 0.0d0, 0.63d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 5V Hybrid #4 (Volts)
#
GD_5VHYB4V_SF = 0.0d0, 1.532d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 5V Hybrid #4 (Amps)
#
```

```
GD_5VHYB4A_SF = 0.0d0, 1.91d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# m5V Hybrid #5 (Volts)
#
GD_M5VHYB5V_SF = 0.0d0, -1.532d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# m5V Hybrid #5 (Amps)
#
GD_M5VHYB5A_SF = 0.0d0, 1.91d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# m5V Hybrid #6 (Volts)
#
GD_M5VHYB6V_SF = 0.0d0, -1.532d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# m5V Hybrid #6 (Amps)
#
GD_M5VHYB6A_SF = 0.0d0, 1.95d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 15V Boost Post Reg (Volts)
#
GD_15VBPR_SF = 0.0d0, 4.08d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# m15V Boost Post Reg (Volts)
#
GD_M15VBPR_SF = 0.0d0, -4.09d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 12V Prim Osc Thermal Control (Amps)
#
GD_12VPOTC_SF = 0.0d0, 0.04d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# 12V Sec Osc Thermal Control (Amps)
#
GD_12VSOTC_SF = 0.0d0, 0.04d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# m2V Discrete Voltage (Volts)
#
GD_M2VDV_SF = 0.0d0, 2.0d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# Hybrid  Heatsink (deg C)
#
GD_HBHS_SF = 0.0d0, 30.2d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# FET Switch Bank Heatsink (deg C)
#
GD_FETSBHS_SF = 0.0d0, 30.2d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# HVPS 0V Reference (Volts)
#
GD_HVPS0VR_SF = 0.0d0, 0.0195d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# HVPS 5V Reference (Volts)
#
GD_HVPS5VR_SF = 0.0d0, 0.0391d0, 0.0d0, 0.0d0, 0.0d0, 0.0d0
#
# Number of bad lidar values within profile that will generate error
#
gi_bad_lid_num = 100
#
# Number of bins to average over for background(4)
```

```
#
gi_bg_bin = 8
#
# 1064 amplifier gain
#
gd_ir_gain = 18.0d0
#
# 1064 volts per count conversion (Volts/cnt)
#
gd_volts_cnt = 0.01560d0
#
# 1064 voltage offset
#
gd_volt_offset = 0.840d0
#
# 1064 detector responsivity factor (Volts/Watt)
#
gd_det_res_ftr = 4.4d7
#
# 1064 Basic time interval - nominally 1 ns
#
d_dthires = 1.0d0
#
# Threshold for computing the received and transmit energy from waveform peak
#
d_thrpcnt = 0.03d0
#
# PRAP Siru Time Tag
#
gd_siru_ttag= 0.0d0, 64.0d0
#
# PRAP Star Tracker #1 Data Latency
#
gd_bst1_datalat = 0.0d0, 64.0d0
#
# PRAP Star Tracker #1 Position X (Currently TBD)
#
gd_bst1_posx = 0.0d0, 1.0d0
#
# PRAP Star Tracker #1 Position Y (Currently TBD)
#
gd_bst1_posy = 0.0d0, 1.0d0
#
# PRAP Star Tracker #1 Intensity
#
gd_bst1_starint = 0.0d0, 0.0625d0
#
# PRAP Star Tracker #1 Position Y (Currently TBD)
#
gd_bst1_posy = 0.0d0, 1.0d0
#
# PRAP Star Tracker #1 CCD Temperature
#
gd_bst1_ccdtemp = 0.0d0, 0.01d0
#
# PRAP Star Tracker #1 BasePlate Temperature
#
gd_bst1_bptemp = 0.0d0, 0.01d0
#
```

```
# PRAP Star Tracker #1 Lens Temperature
#
gd_bst1_lenstemp = 0.0d0, 0.01d0
#
# PRAP Star Tracker #1 8v
#
gd_bst1_8v=0.0d0, 0.1d0
#
# PRAP Star Tracker #1 -9v
#
gd_bst1_n9v=0.0d0, 0.1d0
#
# PRAP Star Tracker #1 4v
#
gd_bst1_4v=0.0d0, 0.1d0
#
# PRAP Star Tracker #1 -5v
#
gd_bst1_n5v=0.0d0, 0.1d0
#
# PRAP Star Tracker #2 Data Latency
#
gd_bst2_datalat = 0.0d0, 64.0d0
#
# PRAP Star Tracker #2 Position X (Currently TBD)
#
gd_bst2_posx = 0.0d0, 1.0d0
#
# PRAP Star Tracker #2 Position Y (Currently TBD)
#
gd_bst2_posy = 0.0d0, 1.0d0
#
# PRAP Star Tracker #2 Intensity
#
gd_bst2_starint = 0.0d0, 0.0625d0
#
# PRAP Star Tracker #2 CCD Temperature
#
gd_bst2_ccdtemp = 0.0d0, 0.01d0
#
# PRAP Star Tracker #2 BasePlate Temperature
#
gd_bst2_bptemp = 0.0d0, 0.01d0
#
# PRAP Star Tracker #2 Lens Temperature
#
gd_bst2_lenstemp = 0.0d0, 0.01d0
#
# PRAP Star Tracker #2 8v
#
gd_bst2_8v=0.0d0, 0.1d0
#
# PRAP Star Tracker #2 -9v
#
gd_bst2_n9v=0.0d0, 0.1d0
#
# PRAP Star Tracker #2 4v
#
gd_bst2_4v=0.0d0, 0.1d0
```

```
#
# PRAP Star Tracker #2 -5v
#
gd_bst2_n5v=0.0d0, 0.1d0
#
# PRAP IST Virtual Tracker Boresight H
#
gd_ist_boreh=0.0d0, 0.5d0
#
# PRAP IST Virtual Tracker Boresight V
#
gd_ist_borev=0.0d0, 0.5d0
#
# PRAP IST Virtual Tracker Focal Length
#
gd_ist_foclngth=0.0d0,1.0d0
#
# PRAP IST Virtual Tracker Time to Center of Integration
#
gd_ist_timcofint=0.0d0, 20.0d0
#
# PRAP LRS Time to Center of Integration
#
gd_lrs_timcofint=0.0d0, 20.0d0
#
# PRAP IST Virtual Tracker Star Magnitude
#
gd_ist_VTstarmag=0.0d0, 0.1d0
#
# PRAP IST Boresight Column
#
gd_ist_borecol=256.0d0,0.001953d0
#
# PRAP IST Boresight Row
#
gd_ist_borerow=256.0d0,0.001953d0
#
# PRAP IST CCD Temperature
#
gd_ist_ccdtemp=-273.16d0,0.007812d0
#
# PRAP IST Lens Cell Temperature
#
gd_ist_lenscellt=-273.16d0,0.007812d0
#
# PRAP LRS Virtual Tracker 0 Centroid Row
#
gd_lrs_VTcentr=0.0d0,0.0078125d0
#
# PRAP LRS Virtual Tracker 0 Centroid Column
#
gd_lrs_VTcentc=0.0d0,0.0078d0
#
# PRAP LRS Integration Time
#
gd_lrs_inttime=0.d00, 1.0d0
#
# PRAP LRS Health CCD Temperature
#
```

```
gd_lrs_hlth_ccdtemp=-273.16, 0.007812
#
# PRAP LRS Health Lens Cell Temperature
#
gd_lrs_hlth_celltemp=-273.16, 0.007812
#
# PRAP ADCS Attitude Q1
#
gd_adcs_att_q1=0.0d0,3.2d-5
#
# PRAP ADCS Attitude Q2
#
gd_adcs_att_q2=0.0d0,3.2d-5
#
# PRAP ADCS Attitude Q3
#
gd_adcs_att_q3=0.0d0,3.2d-5
#
# PRAP ADCS Attitude Q4
#
gd_adcs_att_q4=0.0d0,3.2d-5
#
# PRAP ADCS Orbital Pos X
#
gd_adcs_pos_x=0.0d0,0.25d0
#
# PRAP ADCS Orbital Pos Y
#
gd_adcs_pos_y=0.0d0,0.25d0
#
# PRAP ADCS Orbital Pos Y
#
gd_adcs_pos_z=0.0d0,0.25d0
#
# PRAP ADCS Orbital Vel X
#
gd_adcs_vel_x=0.0d0, 3.2d-4
#
# PRAP ADCS Orbital Vel Y
#
gd_adcs_vel_y=0.0d0, 3.2d-4
#
# PRAP ADCS Orbital Vel Z
#
gd_adcs_vel_z=0.0d0, 3.2d-4
#
# PRAP ADCS Calculated SA 1 Position
#
gd_adcs_cal_pos1=0.0d0,2.0d-4
#
# PRAP ADCS Calculated SA 2 Position
#
gd_adcs_cal_pos2=0.0d0,2.0d-4
#
# Quality Assurance dumping intervals [seconds]
#
I_QA01_DUMP_TME = 16
I_QA02_DUMP_TME = 16
I_QA03_DUMP_TME = 16
```

```
#
# Quality Assurance dumping interval [seconds]
#
D_QA_NRG_HIST = 0.01d0, 0.02d0, 0.03d0, 0.04d0
```

B.3.3 Elevation Constants

```
ELEV_VERS = ANC07 Elev Globals V3.0 October 2002
#
# gravity effect in milligals
#
gravEffTide = 9.8d0
#
# Constants used in calc of earth tide
#
earthTdH2 = 0.609d0
#
# Constants used in calc of earth tide
#
earthTdH3 = 0.291d0
#
# Receiver Impulse Width (in ns)
#
REC_IMPULSE =  4.0d-09
#
# Beam Divergence Angle
#
DIVANGLE =  0.00011d0
#
# QAP Track length (in secs)
#
QAPTRKLEN =  16.0d0
```

B.3.4 Waveform Constants

```
WF_VERS = ANC07 wf V3.0 October 2002
#
# Don't process WF if saturation >= d_psat_stop1
#
D_PSAT_STOP1 = 0.9d0
#
# Don't process WF if saturation >= d_psat_stop2
#
D_PSAT_STOP2 = 0.8d0
#
# If d_psat_spec1 <= WF sat < d_psat_stop1 then use special processing
#
D_PSAT_SPEC1 = 0.2d0
#
# If d_psat_spec2 <= WF sat < d_psat_stop2 then use special processing
#
D_PSAT_SPEC2 = 0.25d0
#
# "If i_slctRegn1 is set, indexSignalBegin1 = indexSignalBegin1 - i_offsetb1"
#
I_OFFSETB1 = 12
#
# "If i_slctRegn2 is set, indexSignalBegin2 = indexSignalBegin2 - i_offsetb2"
#
I_OFFSETB2 = 10
```

```
#
# "If i_slctRegn1 is set, indexSignalEnd1 = indexSignalEnd1 + i_offsete1"
#
I_OFFSETE1 = 12
#
# "If i_slctRegn2 is set, indexSignalEnd2 = indexSignalEnd2 + i_offsete2"
#
I_OFFSETE2 = 10
#
# Max number of iterations during functional fit
#
I_MAXITER1 = 12
#
# Max number of iterations during functional fit
#
I_MAXITER2 = 12
#
# Max number of peaks to fit - land parameters
#
I_MAXFIT1 = 6
#
# Max number of peaks to fit - other than land parameters
#
I_MAXFIT2 = 6
#
# Min peak amplitude = d_Npeak_min1 * (sDevNoise)+ noise
#
D_NPEAK_MIN1 = 4.5d0
#
#
#
D_NPEAK_MIN2 = 4.5d0
#
# Min peak amplitude = d_minAmpPcnt1 * MaxAmp -  used for transmit pulse signal
#     check - note not really % but a fraction
D_MINAMPPCNT1 = 0.10d0
# Min peak amplitude = d_minAmpPcnt2 * MaxAmp -
#
D_MINAMPPCNT2 = 0.12d0
#
# Noise threshhold = d_nsig1 * sDevNoise
#
D_NSIG1 = 4.5d0
#
# Noise threshhold = d_nsig1 * sDevNoise
#
D_NSIG2 = 4.5d0
#
# Min interval between peaks
#
D_INTV_MIN1 = 5.0d0
#
# Min interval between peaks
#
D_INTV_MIN2 = 5.0d0
#
# "0==use all gates, 1==use selected region"
#
I_SLCTREGN1 = 0
```

```
#
# "0==use all gates, 1==use selected region"
#
I_SLCTREGN2 = 0
#
# Min filter width used by W_Smooth1
#
D_FLTRWDMIN = 4.0d0
#
# Max filter width used by W_Smooth1
#
D_FLTRWDMAX = 128.0d0
#
# Time between gates for highest resolution
#
D_DTHIRES = 1.0d0
#
# "(0,1) => weight peaks by area when combining"
#
D_CMB1 = 0.0d0
#
# "(1,0) => use straight average when combining peaks"
#
D_CMB2 = 1.0d0
#
# Min peak sigma
#
D_SIGMAMINIT = 0.5d0
#
# Max peak sigma
#
D_MAXSIGMA = 1000.0d0
#
# Max peak amplitude
#
D_MAXAMP = 300.0d0
#
#
#
D_DNOISE1 = 0.02d0
D_DNOISE2 = 0.02d0
#
#
#
D_DAMPCK1 = 0.02d0
D_DAMPCK2 = 0.02d0
#
#
#
D_DTMCK1 = 0.02d0
D_DTMCK2 = 0.02d0
#
#
#
D_DSGMCK1 = 0.02d0
D_DSGMCK2 = 0.02d0
#
# Min number of iterations during functional fit
#
```

```
I_MINITER = 3
#
# Number of gates to use to calculate noise
#
INGATES2USE1 = 20
INGATES2USE2 = 20
#
# Min number of gates to use to calculate noise
#
IMINNG2USE = 10
#
# "0==use observed noise & sDevNoise, 1==calculate noise & sDevNoise"
#
I_NSCAL1 = 1
I_NSCAL2 = 1
#
# Min number of maxAmp in a row to count as saturated
#
I_MIN4SAT = 3
#
# retracker threshhold = d_thresh_lvl * maxSmoothAmp(WF)
#
D_THRESH_LVL1 = 0.11d0
D_THRESH_LVL2 = 0.15d0
#
# a priori fit-matrix-sigma for noise = 1/SQRT(d_V0ns)
#
D_V0NS1 = 5.0d0
D_V0NS2 = 5.0d0
#
# a priori fit-matrix-sigma for amplitude = 1/SQRT(d_V0amp)
#
D_V0AMP1 = 5.0d0
D_V0AMP2 = 5.0d0
#
# a priori fit-matrix-sigma for location = 1/SQRT(d_V0loc)
#
D_V0LOC1 = 5.0d0
D_V0LOC2 = 5.0d0
#
# a priori fit-matrix-sigma for peak-sigma = 1/SQRT(d_V0sgm)
#
D_V0SGM1 = 5.0d0
D_V0SGM2 = 5.0d0
#
# Minimum noise
#
D_MINNOISE = 0.5d0
#
# Area of telescope
#
D_AREATELE = 0.709d0
#
# Optics transmission
#
D_OPTTRANS = 0.555d0
#
# QA along-track dumping interval time [seconds]
#
```

```
I_QA_DUMP_TME = 16
#
```

B.3.5 Atmosphere Constants

```
ATM_VERS = ANC07 atm Globals V3.0 October 2002
#
# Aerosol scaling factor for 20 sec search
# It seems that 4.0 is good for daytime data, but 5.0 is better for night
#
GD_AER_20S_FTR  = 5.0d0
#
# Aerosol scaling factor for 4 sec search
#
GD_AER_4S_FTR  = 0.3d0
#
# Aerosol threshold factors for new (V2) molecular threshold at 20 sec
#
GD_AER_FTR1 = 3.0d0
GD_AER_FTR2 = 1.2d0
GD_AER_FTR3 = 1.2d0
#
# The two-way aerosol transmission from the top of the atm to the two calibration heights
# for 532 and the one calibration height for 1064
#
GD_AEROSOL_TRANS_HI = 1.0d0
GD_AEROSOL_TRANS_IR = 0.98d0
GD_AEROSOL_TRANS_LO = 0.98d0
#
# Factor to multiply molecular difference for bot threshold for 20 s algm
#
GD_BMDIFF_FTR = 0.98d0
#
# Factor to multiply top threshold for bot threshold for 20 s algm
#
GD_BTHR_FTR = 0.96d0
#
# Calibration cloud factor for determining cloud threshold
#
GD_CAL_CLD_FTR = 1.5d0
#
# Percent of good records needed (in decimal form) for cal cof sums to be
# averaged and output to ancillary file
#
GD_CAL_PCT = 0.5d0
#
# Number of standard deviations to eliminate calibration coefficients
#
GD_CC_NUM_STD = 2.0d0
#
# Use previous granule data if last time is within this number secs of
# current granule time
#
GD_CC_PREV_SEC = 3600.0d0
#
# Ratio to eliminate calibration coefficients
#
GD_CC_RATIO = 0.50
#
```

```
# The height (m) above which clouds detected by the calibration utility are
# considered to be cirrus and the 1064 and 532 data are written to anc file 44
#
GD_CIRRUS_HT = 10000.0
#
# Cloud density threshold
#
GD_CLD_DENSE = 25.0d0
#
# Cloud separation height for using low and high consec values (m)
#
GD_CLD_SEP_HT = 5.0d3
#
# Cloud thickness threshold (m)
#
GD_CLD_THICK =  230.0d0
#
# Cloud threshold factor
#
GD_CLD_THR_FTR =  3.0d0
#
# Critical ratio threshold for 20 s algm
#
GD_CRAT_THR20 = 1.10d1
#
# Critical ratio threshold for 4 s algm
#
GD_CRAT_THR4 = 1.25d0
#
# Upper limit for valid DEM (meters)
#
GD_DEM_HI  =  10000.0d0
#
# Lower limit for valid DEM (meters)
#
GD_DEM_LO  =  -2000.0d0
#
# GLA02%d_DEMmin must be offset by this amount (m)
#
GD_DEM_OFFSET  =  1000.0d0
#
# Cloud/aerosol discrimination thresholds
# from 21 km to ground by 1 km increments
#
#GD_DISCRIM_THR = 6.0d-13, 6.0d-13, 6.0d-13, 6.0d-13, 6.0d-13, 6.0d-13, 6.0d-13, 6.0d-
13, 6.0d-13, 6.0d-13, 6.0d-13, 7.0d-13, 8.0d-13, 1.0d-12, 1.5d-12, 4.0d-12, 6.0d-12,
6.0d-12, 6.0d-12, 6.0d-12, 6.0d-12
GD_DISCRIM_THR = 6.0d-15, 6.0d-15, 6.0d-15, 6.0d-15, 6.0d-15, 6.0d-14, 6.0d-14, 6.0d-
14, 6.0d-14, 6.0d-14, 6.0d-14, 7.0d-14, 8.0d-12, 1.0d-11, 1.5d-10, 4.0d-10, 6.0d-10,
6.0d-10, 6.0d-10, 6.0d-10, 6.0d-10
#
# Diurnal threshold factor
#
GD_DIURNAL_FTR =  0.5d0
#
# Lower threshold for 532 day/night flag
#
GD_DNF_THR1  =  1.0d5
#
```

```
# Upper threshold for 532 day/night flag
#
GD_DNF_THR2  =  1.0d6
#
# 532 background threshold for calibration coefficient elimination (photons/bin)
#
GD_G_BKGD_THR =  4.0d0
#
# Ground detection factor
#
GD_GDET_FTR =  5.0d0
#
# Lab measured 532 calibration coefficient
#
GD_G_LMCALCOF  =  3.0d13
#
# Half width of calibration band around high calibration height (km)
#
GD_HI_HALFWID  =  2.0d0
#
# Lab measured 1064 calibration coefficient
#
GD_IR_LMCALCOF  =  6.0d2
#
# Half width of calibration band around low calibration height (km)
#
GD_LO_HALFWID  =  1.0d0
#
# Factor to multiply average molecular bscs for top threshold for 4 s algm
#
GD_MAER_FTR = 1.1d0
#
# Factor to multiply average molecular bscs for top threshold for 20 s algm
#
GD_MAVG_FTR = 10.0d0
#
# Cloud integrated signal threshold
#
GD_MIN_CLD_SIG  =  5.0d-5
#
# PBL threshold factor
#
GD_PBL_THR_FTR  = 0.40d0
#
# Lower threshold for ratio of GLA07 integrated returns
#
GD_RAT_THR1  =  1.0d-1
#
# Upper threshold for ratio of GLA07 integrated returns
#
GD_RAT_THR2  =  1.0d1
#
# Threshold for range from satellite to geoid (m)
#
GD_RNG_THR = 5.0d5
#
# Total to molecular scattering ratio at the two calibration heights for 532
# and at the one lower height for the 1064 channel
#
```

```
GD_SCAT_RATIO_HI = 1.00d0
GD_SCAT_RATIO_IR = 1.05d0
GD_SCAT_RATIO_LO = 1.03d0
#
# Segment heights for cloud detection (tops of bins)
#
GD_SEG_HTS = 20.5d0, 19.0d0, 17.0d0, 15.0d0, 13.d0, 11.0d0, 10.0d0, 9.0d0, 8.0d0, 7.0d0,
6.0d0, 5.0d0, 4.0d0 3.0d0, 2.0d0, 1.0d0, 0.0d0, -2.0d0, -2.0d0, -2.0d0
#
# Time tolerance for comparing ANC36 and GLA02 times in A_cal_cofs
#
GD_TIME_TOL = 1.0d0
#
# Factor to multiply molecular difference for top threshold for 20 s algm
#
GD_TMDIFF_FTR = 0.98d0
#
# Minimum number of valid 5 Hz profiles in 1 sec buffer
#
GI_1SEC_MIN = 2
#
# Min number of valid 5 Hz profiles in 20 sec buffer
#
GI_20SEC_MIN = 50
#
# Minimum number of valid 1 sec profiles in 20 sec buffer
#
GI_20SEC_MIN1 = 10
#
# Min number of valid 5 Hz profiles in 4 sec buffer
#
GI_4SEC_MIN = 8
#
# Minimum number of valid 1 sec profiles in 4 sec buffer
#
GI_4SEC_MIN1 = 2
#
# Number of bins to constrain the search for the PBL layer at 4 sec
#
GI_4S_LIM_BIN = 12
#
# Number of bins to constrain the search for the PBL layer at 5 Hz
#
GI_5HZ_LIM_BIN = 6
#
# Height (m) above PBL or surface for aerosol search limit
#
GI_AER_BOT_HT = 250
#
# Number of bins above optically thick cloud top to end aerosol search at 4 sec
#
GI_AER_CLD_OFF = 5
#
# Number of bins to constrain the search for aerosol top at 4 sec
#
GI_AER_LIM_BIN = 13
#
# Number of bins for new molecular threshold in 20 sec aerosol detection
#
```

```
GI_AER_NPTS = 450
#
# Minimum number of bins in an aerosol layer in 20 sec algm
#
GI_AER_NUM_BIN = 3
#
# Flag indicating threshold algorithm to use in 20 sec aerosol detection
# 0=old threshold, 1=new molecular threshold
#
GI_AER_THR_F = 0
#
# Lower limit for std for new molecular threshold in 20 sec aerosol detection
#
GI_AER_THR_LL = 40
#
# Upper limit for std for new molecular threshold in 20 sec aerosol detection
#
GI_AER_THR_UL = 160
#
# Start Height (m) for 20 sec aerosol search limit
#
GI_AER_TOP_HT = 36000
#
# Limit for number of bad locations allowed
#
GI_BAD_LOC_LIMIT = 30
#
# Flag to indicate if backscatter quality flags are used to eliminate data
# Used in the PBL routine A_pbl_lay_mod.f90, and possibly elsewhere, but
# definitely only within level 2 software
# Not used = 0, used = 1
#
GI_BSCS_QF_CHK = 0
#
# Flag indicating if 4 sec aerosol layers should be calculated using aerosol
# routine or if aerosol layers found by cloud algorithm should be used.
# 0 = use cloud algorithm, 1 = use aerosol algorithm
#
GI_CALC_AER_F = 1
#
# Start height for cal cof cloud search (m)
#
GI_CAL_CLD_HT = 22000
#
# Number of seconds that cal cof sums are averaged over (segment length)
#
GI_CAL_TIME = 600
#
# Flag to eliminate calibration coefficients (0=no, 1=yes) for data that was taken
# with a background greater than gd_g_bkgd_thr (default vaue of 4.0 photons/bin)
#
GI_CC_ELIM_F = 0
#
# Flag to tell whether to use the high 532 calibration constant or the low
# 1 - Use high calibration constant
# 2 - Use low calibration constant
#
GI_CC_INTERP_F = 1
#
```

```
# This is the order of the least squares polynomial fit that defines a calibration
# constant fo each second of the granule
# -1 - linear piecewise fit between each calibration point
#  0 - 0 order or segment average
#  1 - 1st order or linear least square fit: y = a + bx
#  2 - 2nd order or parobolic fit: y = a + b*x + c*x*x
#  3 - 3rd order or cubic fit: y = a + b*x + c*x*x + d*x*x*x
# ..... etc until a max of 10. Note that GI_CC_MFIT = 10 is essentially a linear piecewise
fit
#
GI_CC_MFIT = -1
#
# Number of points to use in calibration coefficient Interpolation
#
GI_CC_NUM_PTS = 3
#
# Number of bins of 532 and 1064 data to write out when the calibration utility
# detects a cloud. This is for 1064 calibration from cirrus clouds
#
GI_CIRRUS_BINS = 6
#
# Number of consecutive bins for aerosol layer at 20 sec
#
GI_CLD_BIN = 7
#
# Number of consecutive bins for aerosol layer at 20 sec
#
GI_CONS_AER20 = 4
#
# Number of consecutive bins for aerosol layer at 4 sec
#
GI_CONS_AER4 = 4
#
# Number of consecutive bins for ground detection
#
GI_CONS_GRD = 2
#
# Number of consecutive bins for PBL
#
GI_CONS_PBL = 2
#
# Lower threshold for DEM (bins)
#
GI_DEM_THR1 = -13
#
# Upper threshold for DEM (bins)
#
GI_DEM_THR2 = 156
#
# Number of bins above surface to start search for ground detection
#
GI_DET_TOP_BIN = 7
#
# If ground detection is less than DEM by this number of bins, use DEM
# for lower limit of profile search
#
GI_GDET_BIN = 5
#
# Don't add profile to cal cof sums if number invalid bins in profile > this
```

```
# threshold
#
GI_INVALID_THR = 100
#
# Number of consecutive bins for clouds from 10 to 20 km
#
GI_NUM_CLD_HI = 3
#
# Number of consecutive bins for clouds from 0 to 10 km
#
GI_NUM_CLD_LO = 2
#
# Number of consecutive bins for clear sky for 10 to 20 km
#
GI_NUM_CLR_HI = 3
#
# Number of consecutive bins for clear sky for 0 to 10 km
#
GI_NUM_CLR_LO = 2
#
# Backscatter profile is excluded from PBL search if a cloud top is detected
# above this height (m) + 20 sec average PBL height and no cloud bottom or
# ground is detected
# to turn off the elimination of cloud shots, make this larger than 20000
# to enable it, make it about 3000
#
GI_PBL_CLD_HT = 3000
#
# Lower limit for input PBL height
#
GI_PBL_LIM1 = 200
#
# Upper limit for input PBL height
#
GI_PBL_LIM2 = 6000
#
# Height (m) above ground detection for PBL search
#
GI_PBL_TOP_HT = 7000
#
# Troposphere height (m) above average PBL height
#
GI_PBL_TROP_HT = 1000
#
# Height (m) above which polar stratospheric clouds are detected
#
GI_PSC_HT = 14000
#
# Flag indicating if saturated 532 backscatter values should be replaced
# with 1064 backscatter values: Do not replace = 0, replace = 1
#
GI_REPLACE_F = 0
#
# Lower height (m) for random noise calculation
#
GI_RN_BOT = 22000
#
# Upper height (m) for random noise calculation
#
```

```
GI_RN_TOP = 35000
#
# Day number of the year for start of Summer
#
GI_SEASON_DEMARC = 111
#
# Flag to use calculated or lab-measured constant 532 calib coeff; use
# constant value = 0; use calculated value = 1
#
GI_USE_GCC_F  = 1
#
# Flag to use calculated or lab-measured constant 1064 calib coeff; use
# constant value = 0; use calculated value = 1
#
GI_USE_IRCC_F  = 1#
```

Appendix C
GSAS ANC06 File Content

C.1 ANC06 Overview

GSAS software creates ANC06 output files as required by GSDP-31100 (refer to the GLAS Science Software Requirements Document). The ANC06 file contains processing information, error messages, and status messages. The file is in standard GSAS keyword=value format. The format of an ANC06 entry is:

```
[keyword]=[value]
```

The first field [keyword] is a keyword describing the type of information presented. The second field [value] is standardized for use in ANC06. It contains two subfields: the time and a text message. The time is that of the data being processed when the entry was written (if no data have been processed, the time may be 0 or an invalid value). The time is a GSAS-standard time representation (currently UTC seconds).

```
[time] [message]
```

Each type of message may have different message field formats. The rules for parsing an ANC06 entry are:

- the keyword is the string left of the "="
- the data time is the number immediately to the right of the "="
- the messages text is the information after the data time

C.2 ANC06 Example and Description

A sample ANC06 file, created during V1 Acceptance Testing, is examined below. In this example, multiple spaces have been compressed to a single space for readability.

The first entries in the example ANC06 file show what files were initially opened. The message part of each entry is a GSAS standard error/status message. The first field of the error/status message is the error code, the second is the error severity, the third is the reporting routine and the last is the error text.

All initial input files are opened. (this section has been edited for brevity)

```
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
gla05_002_11_0001_0028_2_00_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_02.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_03.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_04.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
```

```
anc01_001_20000101_060000_01_01.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc07_001_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc07_001_01_01.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc07_001_01_03.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc08_001_20000101_000000_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc09_001_20000101_000000_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc12_000_00_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc12_000_00_01.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc13_001_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc16_001_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc17_001_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc24_001_20000101_000000_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc27_001_01_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input) anc27_001_01_01.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
gla06_002_11_0001_0028_2_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
gla12_002_11_0001_0015_0_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
gla13_002_11_0001_0015_0_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
gla14_002_11_0001_0015_0_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
gla15_002_11_0001_0015_0_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
qap06_002_11_0001_0028_2_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
qap12_002_11_0001_0015_0_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
qap13_002_11_0001_0015_0_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
qap14_002_11_0001_0015_0_00_00.dat
         0 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
qap15_002_11_0001_0015_0_00_00.dat
         0 STATUS= 10010, 0, exec_lib, OpenFiles Subroutine status End of execution
```

The control file name is referenced.

```
0 #=---------------------------------------
0 #=Control File Name
0 #=---------------------------------------
0 CF_NAME=cf_p3t1.ctl
```

The control file contents are listed. Notice that "=" is replaced with ":" so there is no chance of the keyword/value processor being confused. Comments are retained. (this section has been edited for brevity)

```
0 CONTROL=#
0 CONTROL=#---------Execution Information
0 CONTROL=#
0 CONTROL= EXEC_KEY:p3t1
0 CONTROL= DATE_GENERATED:21 September 2001
0 CONTROL= OPERATOR:jlee
0 CONTROL=#
0 CONTROL=#----------Static ANC Files
0 CONTROL=#
```

```
        0 CONTROL=# Input ANC07 Files : 00=error, 01=global, 03=atmosphere
        0 CONTROL=#
        0 CONTROL= INPUT_FILE:anc07_001_01_00.dat 0000000 1000000
        0 CONTROL= INPUT_FILE:anc07_001_01_01.dat 0000000 1000000
        0 CONTROL= INPUT_FILE:anc07_001_01_03.dat 0000000 1000000
.
.
.
       0 CONTROL= OUTPUT_FILE:qap14_002_11_0001_0043_0_00_00.dat 84922.3646 90000.0000
       0 CONTROL= OUTPUT_FILE:qap15_002_11_0001_0043_0_00_00.dat 84922.3646 90000.0000
        0 CONTROL=#
        0 CONTROL=#----------Execution Control
        0 CONTROL=#
        0 CONTROL= WAVEFORM_PROCESS:NONE
        0 CONTROL= ELEVATION_PROCESS:ALL
        0 CONTROL= SURFACE_TYPE:ALL
        0 CONTROL=#
        0 CONTROL=#----------End of Control File
        0 CONTROL=#
```

The exec version number is referenced.

```
        0 #=---------------------------------------
        0 #=Exec Version Information
        0 #=---------------------------------------
        0 EXE_VERSION=GLAS_Alt v2.0, October 2001
```

The associated library version numbers are referenced.

```
        0 #=---------------------------------------
        0 #=Common Library Version Information
        0 #=---------------------------------------
        0 LIB_VERSION=libexec V2, October 2001
        0 LIB_VERSION=libplatform V2.0, October 2001
        0 LIB_VERSION=libcntrl V2.0, October 2001
        0 LIB_VERSION=libprod V2.0, October 2001
        0 LIB_VERSION=libfile V2.0, October 2001
        0 LIB_VERSION=libtime V2.0, October 2001
        0 LIB_VERSION=liberr V2.0, October 2001
        0 LIB_VERSION=libanc V2.0, October 2001
        0 LIB_VERSION=libmath V2.0, October 2001
        0 LIB_VERSION=libgeo V2.0, October 2001
```

The associated subsystem version number is referenced.

```
        0 #=---------------------------------------
        0 #=Subsystem Version Information
        0 #=---------------------------------------
        0 WF_VERSION=libwf v2.0, October 2001
        0 ELEV_VERSION=libelev v2.0, October 2001
```

Ancillary files which remain resident in core are read are then closed.

```
        0 STATUS= 10011, 0, ReadAnc, ReadAnc Subroutine status Start of execution
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_00.dat
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_02.dat
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_04.dat
```

```
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_03.dat
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_060000_01_01.dat
        0 STATUS= 10007, 0, ReadANC, Finished reading file:
anc01_001_20000101_060000_01_00.dat
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_00.dat
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_02.dat
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_04.dat
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_03.dat
        0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_01.dat
        0 STATUS= 10007, 0, ReadANC, Finished reading file:
anc01_001_20000101_120000_01_00.dat
        0 STATUS= 10007, 0, ReadANC, Finished reading file: anc07_001_01_00.dat
        0 STATUS= 10006, 0, CloseFile, Closed file: anc07_001_01_00.dat
        0 STATUS= 10007, 0, ReadANC, Finished reading file: anc07_001_01_01.dat
        0 STATUS= 10006, 0, CloseFile, Closed file: anc07_001_01_01.dat
        0 STATUS= 10007, 0, ReadANC, Finished reading file: anc07_001_01_03.dat
        0 STATUS= 10006, 0, CloseFile, Closed file: anc07_001_01_03.dat
        0 STATUS= 10015, 0, ReadANC, Initialized file information
anc08_001_20000101_000000_01_00.dat
        0 STATUS= 10015, 0, ReadANC, Initialized file information anc12_000_00_00.dat
        0 STATUS= 10007, 0, ReadANC, Finished reading file: anc12_000_00_01.dat
        0 STATUS= 10006, 0, CloseFile, Closed file: anc12_000_00_01.dat
        0 STATUS= 50005, 0, C_LoadGeoid, IN E_GETGEOID subroutine status
        0 STATUS= 10007, 0, ReadANC, Finished reading file: anc13_001_01_00.dat
        0 STATUS= 10006, 0, CloseFile, Closed file: anc13_001_01_00.dat
        0 STATUS= 10015, 0, ReadANC, Initialized file information anc16_001_01_00.dat
        0 STATUS= 10015, 0, ReadANC, Initialized file information anc17_001_01_00.dat
        0 STATUS= 10015, 0, ReadANC, Initialized file information
anc24_001_20000101_000000_01_00.dat
        0 STATUS= 10006, 0, CloseFile, Closed file: anc27_001_01_01.dat
        0 STATUS= 10011, 0, ReadAnc, ReadAnc Subroutine status End of execution
```

Parsed execution flag values are referenced.

```
0 #=---------------------------------------
0 #=Execution Control Flags
0 #=---------------------------------------
0 SURFACE_TYPE=Land = T
0 SURFACE_TYPE=Ocean = T
0 SURFACE_TYPE=Sea_Ice = T
0 SURFACE_TYPE=Ice_Sheet = T
0 MANAGER_EXEC=Waveform = F
0 MANAGER_EXEC=Elevation = T
0 ELEVATION_PROCESS=E_CalcLoadTD = T
0 ELEVATION_PROCESS=E_CalcOceanTD = T
0 ELEVATION_PROCESS=E_CalcEarthTD = T
0 ELEVATION_PROCESS=E_CalcPoleTD = T
0 ELEVATION_PROCESS=E_GetGeoid = T
0 ELEVATION_PROCESS=E_CalcTrop = T
0 ELEVATION_PROCESS=E_IntrpPOD = T
0 ELEVATION_PROCESS=E_CalcStdIR = T
0 ELEVATION_PROCESS=E_CalcLdIR = T
```

```
0 ELEVATION_PROCESS=E_CalcOcIR = T
0 ELEVATION_PROCESS=E_CalcSiIR = T
0 ELEVATION_PROCESS=E_CalcIsIR = T
0 ELEVATION_PROCESS=E_CalcSpLoc = T
0 ELEVATION_PROCESS=E_AtmQF = T
0 ELEVATION_PROCESS=E_CalcSlope = T
0 ELEVATION_PROCESS=E_CalcRefl = T
0 ELEVATION_PROCESS=E_ChckReg = T
0 ELEVATION_PROCESS=E_CalcRegRng = T
0 ELEVATION_PROCESS=E_CalcRegParm = T
0 ELEVATION_PROCESS=E_CalcDEM = T
```

Version information from applicable ancillary files is referenced.

```
0 #=----------------------------------------
0 #=Ancillary File Version Information
0 #=----------------------------------------
0 ANC_VERSION=ANC07 Error V2.0 October 2001
0 ANC_VERSION=ANC07 Status V2.0 October 2001
0 ANC_VERSION=ANC07 Globals v2.0 October 2001
0 ANC_VERSION=no ANC07 Atm globals read
0 ANC_VERSION=ANC07 Elev Globals v1.0 Fri Mar 10 11:2
0 ANC_VERSION=no ANC07 WF globals read
0 ANC_VERSION=no ANC07 L1A globals read
0 ANC_VERSION=
0 ANC_VERSION=Version 1.0, 04/12/2000*
0 ANC_VERSION=anc13_dem not read
0 ANC_VERSION=anc16_ltide not read
0 ANC_VERSION=anc17_otide not read
```

Execution trace proceeds. Granules are opened/closed dynamically. (this section has been edited for brevity)

```
0 #=----------------------------------------
0 #=Execution Status
0 #=----------------------------------------
0 STATUS= 10012, 0, ReadData, ReadData Subroutine status
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_00.dat
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_02.dat
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_04.dat
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_03.dat
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_120000_01_01.dat
0 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000101_060000_01_00.dat
0 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000101_060000_01_02.dat
0 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000101_060000_01_04.dat
0 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000101_060000_01_03.dat
0 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000101_060000_01_01.dat
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_180000_01_00.dat
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_180000_01_02.dat
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_180000_01_04.dat
0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
```

```
anc01_001_20000101_180000_01_03.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
anc01_001_20000101_180000_01_01.dat
         0 STATUS= 50006, 0, C_IntrpPOD, IN C_IntrpPOD subroutine status
         0 STATUS= 50008, 0, C_CalcSpLoc, IN C_CALCSPLOC subroutine status
         0 STATUS= 50002, 0, E_CalcOceanTD, IN E_CalcOceanTd subroutine status
         0 STATUS= 50003, 0, E_CalcEarthTD, IN E_CalcEarthTd subroutine status
         0 STATUS= 50001, 0, E_CalcLoadTD, In E_CalcLoadTd subroutine status
         0 STATUS= 50013, 0, E_CalcTrop, IN E_CalcTrop subroutine status
         0 STATUS= 50017, 0, E_CalcDEM, IN E_CalcDEM subroutine status
         0 STATUS= 50010, 0, E_CalcSlope, IN E_CALCSLOPE subroutine status
        0 ERROR=-50022, 2, e_calcslope, Cannot take sq. root of negative quantity cant
take square root of a negative quantity
         0 STATUS= 50021, 0, E_OceanParm, IN E_OceanParm subroutine status
.
.
.
         0 STATUS= 10007, 0, NextGranule, Finished reading file:
gla05_002_11_0001_0028_2_00_00.dat
        0 STATUS= 10006, 0, CloseFile, Closed file: gla05_002_11_0001_0028_2_00_00.dat
         0 STATUS= 10005, 0, OpenFInFile, Opened file: (Input)
gla05_002_11_0001_0028_3_00_00.dat
      8100 STATUS= 10006, 0, CloseFile, Closed file: gla06_002_11_0001_0028_2_00_00.dat
      8100 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
gla06_002_11_0001_0028_3_00_00.dat
      8100 STATUS= 10005, 0, OpenFiles, Opened file: (In-
put)gla06_002_11_0001_0028_3_00_00.dat
      8100 STATUS= 10006, 0, CloseFile, Closed file: qap06_002_11_0001_0028_2_00_00.dat
      8100 STATUS= 10005, 0, OpenFOutFile, Opened file: (Output)
qap06_002_11_0001_0028_3_00_00.dat
      8100 STATUS= 10005, 0, OpenFiles, Opened file: (In-
put)qap06_002_11_0001_0028_3_00_00.dat
.
.
.
   899590 STATUS= 50013, 0, E_CalcTrop, IN E_CalcTrop subroutine status
    899790 STATUS= 50017, 0, E_CalcDEM, IN E_CalcDEM subroutine status
    899800 STATUS= 50010, 0, E_CalcSlope, IN E_CALCSLOPE subroutine status
         0 STATUS= 10007, 0, NextGranule, Finished reading file:
gla05_002_11_0001_0043_4_00_00.dat
        0 STATUS= 10006, 0, CloseFile, Closed file: gla05_002_11_0001_0043_4_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: gla06_002_11_0001_0043_4_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: qap06_002_11_0001_0043_4_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: gla12_002_11_0001_0043_0_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: gla13_002_11_0001_0043_0_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: gla14_002_11_0001_0043_0_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: gla15_002_11_0001_0043_0_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: qap12_002_11_0001_0043_0_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: qap13_002_11_0001_0043_0_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: qap14_002_11_0001_0043_0_00_00.dat
    899990 STATUS= 10006, 0, CloseFile, Closed file: qap15_002_11_0001_0043_0_00_00.dat
    899990 STATUS= 10013, 0, exec_lib, MainWrap Subroutine status Start of execution
```

An summary of input files/records is created. (this section has been edited for brevity)

```
    899990 STATUS= 10013, 0, exec_lib, MainWrap Subroutine status Start of execution
    899990 #=--------------------------------------
```

```
899990 #=Input/Output Wrapup Summary
899990 #=----------------------------------------
899990 INPUT_SUMMARY=gla05_002_11_0001_0028_2_00_00.dat: read 810 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0028_3_00_00.dat: read 1287 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0028_4_00_00.dat: read 1616 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0029_1_00_00.dat: read 1281 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0029_2_00_00.dat: read 1616 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0029_3_00_00.dat: read 1287 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0029_4_00_00.dat: read 1617 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0030_1_00_00.dat: read 1281 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0030_2_00_00.dat: read 1616 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0030_3_00_00.dat: read 1287 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0030_4_00_00.dat: read 1617 records
899990 INPUT_SUMMARY=gla05_002_11_0001_0031_1_00_00.dat: read 1280 records
.
.
.
899990 INPUT_SUMMARY=anc17_001_01_00.dat: read 0 records
899990 INPUT_SUMMARY=anc24_001_20000101_000000_01_00.dat: read 0 records
899990 INPUT_SUMMARY=anc27_001_01_00.dat: read 0 records
899990 INPUT_SUMMARY=anc27_001_01_01.dat: read 0 records
```

An summary of output files/records is created. (this section has been edited for brevity)

```
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0028_2_00_00.dat: wrote 811 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0028_3_00_00.dat: wrote 1288 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0028_4_00_00.dat: wrote 1617 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0029_1_00_00.dat: wrote 1282 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0029_2_00_00.dat: wrote 1617 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0029_3_00_00.dat: wrote 1288 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0029_4_00_00.dat: wrote 1618 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0030_1_00_00.dat: wrote 1282 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0030_2_00_00.dat: wrote 1617 records
899990 OUTPUT_SUMMARY=gla06_002_11_0001_0030_3_00_00.dat: wrote 1288 records
.
.
.
899990 OUTPUT_SUMMARY=qap14_002_11_0001_0029_0_00_00.dat: wrote 0 records
899990 OUTPUT_SUMMARY=qap14_002_11_0001_0043_0_00_00.dat: wrote 0 records
899990 OUTPUT_SUMMARY=qap15_002_11_0001_0015_0_00_00.dat: wrote 0 records
899990 OUTPUT_SUMMARY=qap15_002_11_0001_0029_0_00_00.dat: wrote 0 records
899990 OUTPUT_SUMMARY=qap15_002_11_0001_0043_0_00_00.dat: wrote 0 records
```

An summary of error/status messages is created. (this section has been edited for brevity)

```
899990 #=----------------------------------------
899990 #=Status/Error Wrapup Summary
899990 #=----------------------------------------
899990 ERROR_SUMMARY= 2833426 Cannot take sq. root of negative quantity
899990 STATUS_SUMMARY= 356 Opened file:
899990 STATUS_SUMMARY= 244 Closed file:
899990 STATUS_SUMMARY= 68 Finished reading file:
899990 STATUS_SUMMARY= 1 ReadAnc Subroutine status
899990 STATUS_SUMMARY= 89693 ReadData Subroutine status
899990 STATUS_SUMMARY= 1 MainWrap Subroutine status
899990 STATUS_SUMMARY= 5 Initialized file information
899990 STATUS_SUMMARY= 358731 In E_CalcLoadTd subroutine status
```

```
899990 STATUS_SUMMARY= 179370 IN E_CalcOceanTd subroutine status
899990 STATUS_SUMMARY= 179370 IN E_CalcEarthTd subroutine status
899990 STATUS_SUMMARY= 179371 IN E_GETGEOID subroutine status
899990 STATUS_SUMMARY= 3587400 IN C_IntrpPOD subroutine status
899990 STATUS_SUMMARY= 13915989 IN C_CALCSPLOC subroutine status
899990 STATUS_SUMMARY= 2838356 IN E_CALCSLOPE subroutine status
899990 STATUS_SUMMARY= 3587215 IN E_CalcTrop subroutine status
899990 STATUS_SUMMARY= 8425624 IN E_CalcDEM subroutine status
899990 STATUS_SUMMARY= 1024984 IN E_LandParm subroutine status
899990 STATUS_SUMMARY= 2557848 IN E_OceanParm subroutine status
```

Any remaining open files are closed.

```
899990 STATUS= 10013, 0, exec_lib, MainWrap Subroutine status End of execution
899990 STATUS= 10014, 0, exec_lib, CloseFiles Subroutine status Start of execution
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_120000_01_00.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_180000_01_00.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_120000_01_02.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_180000_01_02.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_120000_01_03.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_180000_01_03.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_120000_01_04.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_180000_01_04.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_120000_01_01.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc01_001_20000102_180000_01_01.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc08_001_20000101_000000_01_00.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc09_001_20000101_000000_01_00.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc12_000_00_00.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc16_001_01_00.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc17_001_01_00.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc24_001_20000101_000000_01_00.dat
899990 STATUS= 10006, 0, CloseFile, Closed file: anc27_001_01_00.dat
899990 STATUS= 10014, 0, exec_lib, CloseFiles Subroutine status End of execution
```

Appendix D
GSAS Error and Status Codes

This section list errors and possible solutions for each error defined in GSAS.

D.1 General Errors

-10001 Error Opening File for Input:
The operating system was unable to open the specified input file. Verify that the requested file exists in (or is correctly linked into) the current directory. Verify the spelling of the filename within the control file (filenames are case-sensitive). Verify that the user has read permission on the file.

-10002 No control file specified.
A control file must be specified as a command-line argument when GLAS_Exec is executed. If a control file was specified, then perform the following verification steps: Verify that the control file exists in (or is correctly linked into) the current directory. Verify that the user has read permission on the control file

-10003 Bad rec_ndx in control file.
A bad start or stop time was specified in the control file. The start/stop times are fractional numbers representing UTC seconds. The times must contain only digits and the decimal point.

-10004 Stop rec_ndx > start rec_ndx.
A start time which was chronologically greater than the stop time was found in the control file. Start times must be less than stop times.

-10005 Previous granule stop > current start.
Multiple granules of the same type were specified in the control file. The start time of a previous granule was chronologically greater than the stop time of the current granules. Multiple granules of the same type must not have time spans which overlap within the same control file. The granules must be entered in the control file in chronological order.

-10006 Error Opening File for Output:
The operating system was unable to open the specified output file. Verify that the requested file does <u>not</u> already exists in (or is correctly linked into) the current directory. Verify that the user has write permission on the current directory.

-10007 Error Closing File:
The operating system returned an error code when attempting to close a file. Verify that opened files were not moved from the current directory while GLAS_Exec was running. This is an unusual error and would normally indicate a programming error.

-10008 Error Reading File:
The operating system returned an error code when attempting to read a file. Verify that files listed in the control file are present and accessible in the current directory.

This is an unusual error and would normally indicate a programming error or a corrupt input file.

-10009 Error Writing File:
The operating system returned an error code when attempting to write a file. Verify that files listed in the control file are present and accessible in the current directory. This is an unusual error and would normally indicate a programming error or a full file system.

-10010 Error Reading ANC07:
The operating system returned an error code when attempting to read the specified ANC07 file. Verify that the ANC07 files listed in the control file are present and accessible in the current directory. If all files are present, this indicates corruption in the ANC07 file. Appendix B describes the format of ANC07 files. verify the ANC07 file contents.

-10011 Unknown Keyword in ANC07:
The ANC07 file reader has found a keyword it does not recognize in the ANC07 file. This usually indicates corruption in the ANC07 file. Appendix B lists recognized ANC07 keywords. Verify the ANC07 file contents.

-10012 Multiple single-instance keywords:
The control file parser has detected multiple instances of single-instance keywords in the control file. Appendix C details single and multiple-instance keywords in the control file. Verify the control file contents.

-10013 Multiple-instance keyword limit exceeded:
The control file parser has detected that the number of instances of a multiple-instance keyword has been exceeded. Verify the control file contents.

-10014 Unrecognized line in control file:
The control file parser has found a line in the control file which it does not recognize. Appendix C details the format of the control file. Verify the control file contents.

-10015 Unknown value in keyword/value pair:
The control file parser has found a line in the control file with a value it does not recognize. Certain keywords are required to have only certain values. Appendix C lists possible values within the control file. Verify the control file contents.

-10016 I/O Error Opening Control File:
The operating system was unable to open the specified control file. Verify that the requested file exists in (or is correctly linked into) the current directory. Verify the spelling of the filename on the command line. Verify that the user has read permission on the file.

-10017 Error Reading Control File:
The operating system returned an error code when attempting to read the control file. Verify that the control file is present and accessible in the current directory. This is an unusual error and would normally indicate a programming error or a corrupt control file.

-10018 Specified Unknown File Type:
An unknown file type was specified in the control file. A file type is represented by the first 5 characters of the filename. Appendix A lists recognized GLAS file types. Verify the control file.

-10019 GLA01 Unknown Record Type
An unknown record type was found within the GLA01 file. Valid record types are header, body, long and short. This error indicates a corrupt GLA01 file or a program error.

-10020 GLA01 Exceeded Waveform Record Limit
The number of waveform records per second was exceeded in the GLA01 file. This error indicates a corrupt GLA01 file or a program error.

-10021 GLA00 Unknown APid
An unknown APID was detected in the GLA00 data. This error indicates corrupt GLA00 data.

-10022 GLA00 Wrong APid
.This error indicates that the expected APID number was not found in the file being read. This error indicates corrupt GLA00 data.

-10023 Max APIDs per sec exceeded
This indicates that the maximum number of APIDs per second has been exceeded. This error indicates corrupt GLA00 data.

-10024 Data Time < Granule Start Time
This indicates that data was lost because the time of the current record is less than the start time of the current granule. Verify the start and stop times in the control file.

-10025 Error Reading Standard Atmosphere file header
An error was detected reading the Standard Atmosphere file header. This indicates a corrupt Standard Atmosphere file.

-10026 Error Reading Standard Atmosphere file data
An error was detected reading the Standard Atmosphere file data. This indicates a corrupt Standard Atmosphere file.

-10027 Error Reading Global Aerosol map
An error was detected reading the Global Aerosol file data. This indicates a corrupt Global Aerosol file.

-10028 Error Reading Aerosol Troposphere map
An error was detected reading the Aerosol troposphere file data. This indicates a corrupt Aerosol Troposphere map file.

-10029 Duplicate APID found
An error was detected when duplicate APID records were found while processing GLA00 APID data. A duplicate APID is defined as one having the same APID number and same mission elapsed time.

-10030 EOF or Error Reading ANC33
An error was detected while reading the ANC33 file. Verify that the file exists and has the correct format.

-10031 Exceeded maximum GLA00 index entries
The number of GLA00 index entries in the ANC29 file exceeded the maximum allowed. Contact the GSAS Development team.

-10032 Non-zero version in APID Primary Hdr
An inconsistency was detected in an APID primary header field. This is most likely due to a corrupt APID file.

-10033 Mismatched APID numbers in phdr and file
An error was detected when verifying that the APID number within the APID data did not match the expected APID number designated by the file type. This is most likely due to a corrupt APID file.

-10034 Flag indicates no secondary header in APID
An error was detected when checking the secondary header flag within the APID data. This is most likely due to a corrupt APID file.

-10035 Gap in APID sequence count
A gap was detected in the APID sequence count. This most likely indicates missing data or a corrupt APID file.

-10036 Mismatch in APID lengths
A mismatch was detected when verifying the APID length field within the secondary header against the expected APID length, based on the file type of the APID. This is most likely due to a corrupt APID file.

-10037 GPS Delta time exceeds limit
The time between consecutive GPS updates exceeds a preset limit. This is most likely due to missing data.

-10038 Inconsistent APID shot count
The APID shot counts are inconsistent. This indicates that the APID alignment process may not work correctly. Contact the GSAS Development Team.

-10039 Overflowed number of reserved header records
This indicates the number of product header records has exceeded the expected number of product header records. The product header records may be incomplete. Contact the GSAS Development Team.

-10040 Reference Orbit File is empty
This indicates the reference orbit file is corrupt.

-10041 Did not find desired rev in rev file
This indicates that data time is not within the predicted orbit file.

-**10042 Granule time not within Ref Orb file**
This indicates that data time is not accounted for in the reference orbit id table.

-10043 Undefined Segment type
This indicates an error in routine while determining segment type.

-10044 Error running CreateGran_util
This indicates that the utility terminated abnormally. An error code will be generated where the error occurred.

-10045 Time error: current < previous
This error indicates that time has moved backwards. This indicates a corrupt APID or product file.

-10046 Error returned from script
An error was returned from the SDMS_met_script. This is most likely due to a corrupt ANC40 file.

-10047 File naming conventions do not match
met_util has detected that the file naming conventions do not match when comparing the names of the ANC40 and ANC01 files.

-10048 Unknown header keyword
This indicates that the processing has detected a header keyword which was not expected. This may indicate lack of the appropriate input ANC45 file or a corrupted product file.

-10049 APID record edited. Shot count > limit
This indicates that a sanity check of the APID has detected a shot counter whose value is out of the nominal range (0-200). This normally indicates bad APID data. The APID record in question was removed from processing.

D.1.1 L1A Errors

-20001 Error reading PAD data Eng data
No Used.

-20002 Error reading PAD data Eng data
Not Used.

-20003 Background value out of range
If background value is outside of threshold limits, lidar values are set to invalid and lidar quality flag is set to 3.

-20004 Energy value out of range
If transmitted energy value is less than threshold value, lidar values are set to invalid.

-20005 Large number bad lidar values within profile
If the number of lidar values, within a profile, that fall outside of threshold limits exceeds a certain number, then the lidar values are set to invalid.

-20006 APID not available - no atm processed
Either the ancillary science packet data are unavailable or both the photon science packet and the cloud digitizer packet data are unavailable for the record. No L1A atmosphere data are processed and invalid values are written to GLA02.

-20007 APID not available - atm processed
Either the photon science packet data or the cloud digitizer packet data are unavailable for the record. Available L1A atmosphere data are processed. Unavailable values given invalid values on GLA02.

-20008 Range value out of range
If range to start of profile value is outside of threshold limits, a default value is used.

-20009 Attenuation setting <= 0, set to 1
If the attenuation setting is less than 1, it is set equal to 1.

-20010 1064 computed transmit energy unavailable
If the 1064 nm transmitted energy is not available, a default value is used.

-20011 No ANC32 data available
No ANC32 data are available for processing. Verify the correctness of the control file.

-20012 No ANC33 data available
No ANC33 data are available for processing. Verify the correctness of the control file.

-20013 No ANC25 data available
No ANC25 data are available for processing. Verify the correctness of the control file.

-20014 Current number of seconds exceeds max number

-20015 Using estimated shot time
Shot time is being estimated instead of calculated. This is most likely due to a missing APID19 record.

-20016 Filter number exceeds limit
This indicates that the filter number in APID19 exceeds the physical number of filters. This indicates bad APID19 data.

-20017 Time inconsistency
This error indicates that the precise time computed in shot_time_mod differs from the estimated time computed using the MET counter by greater than one second. This could be due to bad data or bad ANC33/ANC25 entries.

-20018 No LRS alignment for entire record
This error indicates that an entire record of LRS data could not be aligned with the correct APID19. The LRS data are lost and were not written to GLA04. This error could be caused by missing APID19 data or a timing problem.

-20019 No IST alignment for entire record
This error indicates that an entire record of IST data could not be aligned with the correct APID19. The IST data are lost and were not written to GLA04. This error could be caused by missing APID19 data or a timing problem.

-20020 No LRS alignment for shot
This error indicates that a 1/10 second shot of LRS data could not be aligned with the

correct APID19 shot. The LRS data are lost and were not written to GLA04. This error could be caused by missing APID19 data or a timing problem.

-20021 No IST alignment for shot
This error indicates that a 1/10 second shot of IST data could not be aligned with the correct APID19 shot. The IST data are lost and were not written to GLA04. This error could be caused by missing APID19 data or a timing problem.

-20022 No LRS alignment for image
This error indicates that an LRS image could not be aligned with the correct APID19 shot. The image data are lost and were not written to GLA04. This error could be caused by missing APID19 data or a timing problem.

D.2 Waveform Errors

-30001 Singular Matrix
Matrix inversion performed as part of the least squares fitting of a waveform to a Gaussian has yielded a singular matrix, and no solution is available.

-30002 Number of peaks found greater than max
The number of peaks found compounded within a single returned waveform exceeds the max number allowable.

-30003 No signal
Not used.

-30004 Standard deviation is zero
Not used.

-30005 Bad Frame
Not used.

-30006 Error from C_InterpPOD
Not used.

-30007 Error from C_CalcSpLoc
Not used.

-30008 Error from WF_Mgr
Filter number exceeds known values. It is less than 1 or greater than 6. Probable data error.

-30009 W_Estimates, Underflow creating DEXP values
not used

-30010 WFMgr, Inconsistent Compression Parameters.
Compression parameters are inconsistent. Input parameters have not been defined consistently. Probable error in telemetry.

-30011 GLA05_scal_mod, Overflow in sigma computation.
This implies that the errors of the parameters solved for with the Gaussian fit are not within acceptable bounds.

-30012 QAP file write error on Packet data
Write error encountered while writing a record of 16-second processing packet data to the QAP file. Possible system-related error.

-30013 QAP file write error on Summary data
Write error encountered while writing a record of granule summary data to the QAP file. Possible system-related error.

-30014 W_CalcReflct, Illegal value trans energy pulse.
An unphysical value for the energy of the transmitted pulse exists, and is equal to or less than 0.0 joules. Reflectance is set to default to avoid division by zero and/or an unphysical computation.

-30015 GLA05_scal_mod, Overflow in reflectivity comp
This implies that values used in the reflectivity calculation are outside bounds.

-30016 WFMgr_mod, Indeterminate waveform type
Flag that identifies a waveform as short vs. long cannot be properly retrieved. Probable data error or missing data.

-30017 W_ParamWithFit, d_bg_Noise/d_sDevNoise invalid

D.3 Atmosphere Errors

Subsystem errors and corrective action will be fully documented in V2 of this document.

-40001 DEM out-of-bounds
If the DEM value is invalid or outside of threshold limits, do not process record.

-40002 532 integrated return flag poor or bad
If the 532 nm integrated return flag indicates that the 532 nm lidar data are poor, the 1 Hz backscatter quality flag is set to 1.

-40003 Ratio of integrated returns out-of-bounds
If the ratio of the 532 nm integrated return / molecular integrated return is outside of threshold limits, then the 1 Hz backscatter quality flag is set to 1.

-40004 Large num bad recs not incl in data
Not used.

-40005 Excessive num bad recs not incl in data
Not used.

-40006 Time between recs greater than threshold
Not used.

-40007 Divide by zero
If the denominator value is 0, set it to a default value.

-40008 Exponent too large
If the exponent value exceeds a certain threshold limit, set the exponent to that limit.

-40009 Index beyond gi_cld_lays
Not used.

-40010 Index beyond gi_pb_aer_lays
Not used.

-40011 Deficient 532 laser energy flag
If the 532 nm energy laser flag indicates that the 532 nm transmitted laser energy is deficient, the 532 nm backscatter quality flag is set to 1.

-40012 Deficient 1064 laser energy flag
If the 1064 nm energy laser flag indicates that the 1064 nm transmitted laser energy is deficient, the 1064 nm backscatter quality flag is set to 1.

-40013 Excess num cloudy recs not incl in data
Not used.

-40014 PBL signal ratio less than zero
Not used.

-40015 Input PBL height out-of-bounds
If PBL height value is outside threshold limits, calculate PBL height as height of maximum profile signal.

-40016 Aerosol signal ratio less than zero
Not used.

-40017 Insufficient filtered data for processing
If the number of valid backscatter values averaged over a period of time is below a certain percentage, then data is not processed for that time span. This affects PBL and aerosol detection.

-40018 Random noise not valid for cloud detection
If at least two backscatter values are not available to calculate a standard deviation, then the random noise is set to invalid.

-40019 Too many bad locations
This error is for met interpolation only. If the input latitude or longitude value is invalid, use last saved value. If the number of invalid values exceeds a threshold limit, this error is called.

-40020 PBL QA array index beyond array size
If the PBL height QA array is indexed by a number greater than the array dimension, no more data is added to the array.

-40021 Invalid lat/lon from GLA07
This error is for the atmosphere manager only. If control flags are set to skip POD processing and the latitude or longitude value from GLA07 is invalid, processing stops.

-40022 Invalid DEM from GLA07
This error is for the atmosphere manager only. If control flags are set to skip POD processing and the DEM value from GLA07 is invalid, the DEM value is set to 0.0.

-40023 Input product is not GLA02
Not used.

-40024 Invalid value written to ancillary file
This error is for the atmosphere utility only. If a certain percentage of lidar values within a time segment are invalid, an invalid average is written to the anc36 file for the segment. This message may be seen often if there were clouds present when the data were collected.

-40025 Array index is greater than size allowed
If an array is indexed by a number greater than the array dimension, processing stops. Check the array dimension in the atmosphere constants file and increase if necessary.

-40026 No ANC36 data for current granule
If there is no ANC36 file for the current granule, then atmosphere backscatter processing stops.

-40027 Cal cof avg is invalid after elim tests
If the calibration coefficient is invalid after the elimination tests, use the original average taken before the tests and set the calibration coefficient quality flag to 2.

-40028 ANC36 current granule data is in first structure
ANC36 data for the current granule are expected in the second structure and data for the previous granule are expected in the first structure. If data for the current granule are in the first structure, processing may continue, but data for the previous granule are not available.

-40029 Matrix singularity in polynomial fit routine

-40030 less than 3 cal constants for least sqrs fit

-40031 GLA10_scal_mod, Overflow in aerosol bs compute

-40032 GLA10_scal_mod, Overflow in aerosol prof compute

D.4 Elevation Errors

-50001 Error reading met height file
An error was detected reading the meteorological height file. This indicates a corrupt met height file.

-50002 Error reading met temperature file
An error was detected reading the meteorological temperature file. This indicates a corrupt met temperature file.

-50003 Error reading met relative humidity file
An error was detected reading the meteorological relative humidity file. This indicates a corrupt met relative humidity file.

-50004 Error reading met precip water table file
An error was detected reading the meteorological precip water file. This indicates a corrupt met precip water file.

-50005 Elevation more than max pressure level ht
The elevation is higher than the maximum pressure level height. This indicates elevation is suspect.

-50006 Max Iterations exceeded in ODE calcs.
This indicates that the data is suspect.

-50007 Step size underflow in BS-step method
This indicates that the data is suspect.

-50008 Value of cosine GT 1.0 in Geoloc
The cosine of angle is greater that 1. Most probably due to round-off errors in calculation.

-50009 Value of cosine LT -1.0 in Geoloc
The cosine of angle is greater that 1. Most probably due to round-off errors in calculation.

-50010 Max Iterations exceeded in Geoloc
This occurs when number of iterations exceeds preset limits. This indicates data is suspect.

-50011 POD file structure is empty
This occurs when POD file structure is not loaded from control file. Need to check if control file contains ANC08 or ANC20 input file.

-50012 Time not within vals in POD file structure
This occurs when data time is not within time range in the POD file structure. Need to check if correct start/stop times are entered in control file for the ANC08/ANC20 files.

-50013 Error reading 10 consec recs in POD file
This occurs when data time is past the last time in the ANC08/ANC20 file.

-50014 Error reading Load Tide coeffs file
An error was detected reading the load tide coefficients file. This indicates a corrupt load tide coefficients file.

-50015 Error reading Geoid coeffs file
An error was detected reading the geoid file. This indicates a corrupt geoid file.

-50016 Error reading DEM file
An error was detected reading the DEM file. This indicates a corrupt DEM file.

-50017 Error reading 1 deg Ocean Mask file
An error was detected reading the ocean mask file. This indicates a corrupt ocean mask file.

-50018 Size of ocean td array must be increased
Error in legacy code. Need to refer to Richard Ray's GOT00 documentation.

-50019 Likely error in Check1 of legacy ocean td
Error in legacy code. Need to refer to Richard Ray's GOT00 documentation.

-50020 Lat/Lon out of bounds in legacy ocean td
Error in legacy code. Need to refer to Richard Ray's GOT00 documentation.

-50021 Increase dimensions in call to UTCSRI
Error in legacy code. Need to refer to Richard Ray's GOT00 documentation.

-50022 Cannot take sq. root of negative quantity
This indicates data is suspect.

-50023 Bad Frame flag set in E_SeaIceParm (rem)
Not used.

-50024 Bad Frame flag set in E_LandParm (rem)
Not used.

-50025 Bad Frame flag set in E_OceanParm (rem)
Not used.

-50026 Bad Frame flag set in E_CalcSlope (rem)
Not used.

-50027 Bad Frame flag set in E_CalcRange (rem)
Not used.

-50028 Geopotential Ht >max rel hum ht
The geopotential height is more that the maximum height for relative humidity. The relative humidity is set to zero.

-50029 ROT file structure is empty
This occurs when ROT matrix file structure is not loaded from control file. Need to check if control file contains ANC24 input file.

-50030 Time not within vals in ROT file structur
This occurs when data time is not within time range in the ROT file structure. Need to check if correct start/stop times are entered in control file for the ANC24 files.

-50031 Error reading recs in ROT file
This indicates a corrupt ROT file.

-50032 Invalid input POD value in c_calcsploc
This indicates that data could not be geolocated because the POD value is invalid.

-50033 Error reading fine Surface Type file
An error has occurred reading the ANC27_00 file. This indicates a corrupt 2min x 2min ANC27 file.

-50034 Error reading coarse Surface Type file
An error has occurred reading the ANC27_01 file. This indicates a corrupt 1deg x 1deg ANC27 file.

-50035 LAT/LON out of bounds in coarse surf type
The latitude/longitude is not within the lat/lon boundaries. This indicates that data is suspect.

-50036 LAT/LON out of bounds in fine surf type
The latitude/longitude is not within the lat/lon boundaries. This indicates that data is suspect.

-50037 Fine surface type grid index out of bounds
This indicates that data is suspect.

D.5 HP Runtime Error Codes

This section provides a list of HP runtime error codes. These codes are taken directly from *HP Fortran 90 Programmers Reference; Product Number: B3909DB; Fortran 90 Compiler for HP-UX; Document Number: B3908-90002; October 1998.*

Error No.	Error Message	Description	Action
900	ERROR IN FORMAT	FORMAT statement syntax contains an error.	Refer to the syntax for "FORMAT" on page 330. Also see Chapter 8, "I/O and file handling," on page 171 for the syntax of the format specification and edit descriptors.
901	NEGATIVE UNIT NUMBER SPECIFIED	Unit number was not greater than or equal to zero.	Use a non-negative unit number.
902	FORMATTED I/O ATTEMPTED ON UNFORMATTED FILE	Formatted I/O was attempted on a file opened for unformatted I/O.	Open the file for formatted I/O or perform unformatted I/O on this file.9
903	UNFORMATTED I/O ATTEMPTED ON FORMATTED FILE	Unformatted I/O was attempted on a file opened for formatted I/O.	Open the file for unformatted I/O or perform formatted I/O on this file.
904	DIRECT I/O ATTEMPTED ON SEQUENTIAL FILE	Direct operation attempted on sequential file, direct operation attempted on opened file connected to a terminal.	Use sequential operations on this file, open file for direct access, or do not do direct I/O on a file connected to a terminal.

Error No.	Error Message	Description	Action
905	ERROR IN LIST-DIRECTED READ OF LOGICAL DATA	Found repeat value, but no asterisk. First character after optional decimal point was not T or F.	Change input data to correspond to syntax expected by list-directed input of logicals, or use input statement that corresponds to syntax of input data.
907	ERROR IN LIST-DIRECTED READ OF CHARACTER DATA	Found repeat value, but no asterisk. Characters not delimited by quotation marks.	Change input data to correspond to syntax expected by list-directed input of characters, or use input statement that corresponds to syntax of input data.
908	COULD NOT OPEN FILE SPECIFIED	Tried to open a file that the system would not allow for one of the following reasons: access to the file was denied by the file system due to access restriction; the named file does not exist; or the type of access request is impossible.	Correct the pathname to open the intended file.
909	SEQUENTIAL I/O ATTEMPTED ON DIRECT ACCESS FILE	Attempted a BACKSPACE, REWIND, or ENDFILE on a terminal or other device for which these operations are not defined.	Do not use the BACKSPACE, REWIND, and ENDFILE statements.
910	ACCESS PAST END OF RECORD ATTEMPTED	Tried to do I/O on record of a file past beginning or end of record.	Perform I/O operation within bounds of the record, or increase record length.
912	ERROR IN LIST I/O READ OF COMPLEX DATA	While reading complex data, one of the following problems has occurred: no left parenthesis and no repeat value; repeat value was found but no asterisk; or no closing right parenthesis.	Change input data to correspond to syntax expected by list-directed input of complex numbers, or use input statement corresponding to syntax of input data.

Error No.	Error Message	Description	Action
913	OUT OF FREE SPACE	Library cannot allocate an I/O block (from an OPEN statement), parse array (for formats assembled at run-time), file name string (from OPEN) characters from list-directed read, or file buffer. The program may be trying to overwrite a shared memory segment defined by another process.	Allocate more free space in the heap area, open fewer files, use FORMAT statements in place of assembling formats at run time in character arrays, or reduce the maximum size of file records.
914	ACCESS OF UNCONNECTED UNIT ATTEMPTED	Unit specified in I/O statement has not previously been connected to anything.	Connect unit using the OPEN statement before attempting I/O on it, or perform I/O on another, already connected, unit.
915	READ UNEXPECTED CHARACTER	Read a character that is not admissible for the type of conversion being performed. Input value was too large for the type of the variable.	Remove from input data any characters that are illegal in integers or real numbers.
916	ERROR IN READ OF LOGICAL DATA	An illegal character was read when logical data was expected.	Change input data to correspond to syntax expected when reading logical data or use input statement corresponding to syntax of input data.
917	OPEN WITH NAMED SCRATCH FILE ATTEMPTED	Executed OPEN statement with STATUS='SCRATCH', but also named the file. Scratch files must not be named.	Either remove the FILE= specifier, or open the file with a status other than STATUS='SCRATCH'.
918	OPEN OF EXISTING FILE WITH STATUS='NEW' ATTEMPTED	Executed OPEN statement with STATUS='NEW', but file already exists.	Either remove the STATUS= specifier from the OPEN statement, or use the STATUS='OLD'; STATUS='UNKNOWN'; or STATUS='REPLACE' specifier.
920	OPEN OF FILE CONNECTED TO DIFFERENT UNIT ATTEMPTED	You attempted to open a file that is already open with a different unit number.	Close the file with the current unit number before attempting to open it with a different unit number.

Error No.	Error Message	Description	Action
921	UNFORMATTED OPEN WITH BLANK SPECIFIER ATTEMPTED	OPEN statement specified FORM='UNFORMATTE D' and BLANK=xx.	Either use FORM='FORMATTED' or remove BLANK=xx.
922	READ ON ILLEGAL RECORD ATTEMPTED	Attempted to read a record of a formatted or unformatted direct file that is beyond the current end-of-file.	Read records that are within the bounds of the file.
923	OPEN WITH ILLEGAL FORM SPECIFIER ATTEMPTED	FORM= specified string other than 'FORMATTED' or 'UNFORMATTED'.	Use either 'FORMATTED' or 'UNFORMATTED' for the FORM= specifier in an OPEN statement.
924	CLOSE OF SCRATCH FILE WITH STATUS='KEEP' ATTEMPTED	The file specified in the CLOSE statement was previously opened with 'SCRATCH' specified in the STATUS= specifier.	Open the file with a STATUS= , specifying a string other than 'SCRATCH' or do not specify STATUS='KEEP' in the CLOSE statement for this scratch file.
925	OPEN WITH ILLEGAL STATUS SPECIFIER ATTEMPTED	STATUS= specified string other than 'OLD' 'NEW' 'UNKNOWN' 'REPLACE' or 'SCRATCH'.	Use 'OLD', 'NEW', 'UNKNOWN', 'REPLACE' or 'SCRATCH' for the STATUS= specifier in OPEN statement.
926	CLOSE WITH ILLEGAL STATUS SPECIFIER ATTEMPTED	STATUS= specified string other than 'KEEP' or 'DELETE'.	Use 'KEEP' or 'DELETE' for the STATUS= specifier in a CLOSE statement.
927	OPEN WITH ILLEGAL ACCESS SPECIFIER ATTEMPTED	ACCESS= specified string other than 'SEQUENTIAL' or 'DIRECT'.	Use 'SEQUENTIAL' or 'DIRECT' for the ACCESS= specifier in an OPEN statement.
929	OPEN OF DIRECT FILE WITH NO RECL SPECIFIER ATTEMPTED	OPEN statement has ACCESS='DIRECT', but no RECL= specifier.	Add RECL= specifier to OPEN statement, or specify ACCESS= 'SEQUENTIAL'.
930	OPEN WITH RECL LESS THAN 1 ATTEMPTED	RECL= specifier in OPEN statement was less than or equal to zero.	Specify a positive number for RECL= specifier in OPEN statement.
931	OPEN WITH ILLEGAL BLANK SPECIFIER ATTEMPTED	BLANK= specified string other than 'NULL' or 'ZERO'	Use 'NULL' or 'ZERO' for BLANK= specifier in OPEN statement.

Error No.	Error Message	Description	Action
933	END (OR BEGIN) OF FILE WITH NO END=x SPECIFIER	End-of-file mark read by a READ statement with no END= specifier to indicate label to which to jump.	Use the END= specifier to handle EOF, or check logic.
937	ILLEGAL RECORD NUMBER SPECI-FIED	A record number less than one was specified for direct I/O.	Use record numbers greater than zero.
942	ERROR IN LIST-DIRECTED READ - CHARACTER DATA READ FOR ASSIGN-MENT TO NON-CHARACTER VARIABLE	A character string was read for a numerical or logical variable.	Check input data and input variable type.
944	RECORD TOO LONG IN DIRECT UNFOR-MATTED I/O	Output requested is too long for specified (or pre- existing) record length.	Make the number of bytes output by WRITE less than or equal to the file record size.
945	ERROR IN FORMAT-TED I/O	More bytes of I/O were requested than exist in the cur-rent record.	Match the format to the data record.
953	NO REPEATABLE EDIT DESCRIPTOR IN FORMAT STRING	No format descriptor was found to match I/O list items.	Add at least one repeat-able edit descriptor to the format statement.
956	FILE SYSTEM ERROR	The file system returned an error status during an I/O oper-ation.	See the associated file system error message.
957	FORMAT DESCRIP-TOR INCOMPATIBLE WITH NUMERIC ITEM IN I/O LIST	A numeric item in the I/O list was matched with a non-numeric edit descriptor.	Match format descriptors to I/O list.
958	FORMAT DESCRIP-TOR INCOMPATIBLE WITH CHARACTER ITEM IN I/O LIST	A character item in the I/ O list was matched with an edit descriptor other than A or R.	Match format descriptors to I/O list.
959	FORMAT DESCRIP-TOR INCOMPATIBLE WITH LOGICAL ITEM IN I/O LIST	A logical item in the I/O list was matched with a edit descriptor other than L.	Match format descriptors to I/O list.
973	RECORD LENGTH DIFFERENT IN SUB-SEQUENT OPEN	Record length specified in sec-ond OPEN conflicted with the value as opened.	Only BLANK=, DELIM=, and PAD= specifiers may be changed by a redun-dant OPEN.

Error No.	Error Message	Description	Action
974	RECORD ACCESSED PAST END OF INTERNAL FILE RECORD (VARIABLE)	An attempt was made to transfer more characters than internal file length.	Match READ or WRITE statement with internal file size.
975	ILLEGAL NEW FILE NUMBER REQUESTED IN FSET FUNCTION	The file number requested to be set was not a legal file system file number.	Check that the OPEN succeeded and the file number is correct.
976	UNEXPECTED CHARACTER IN "NAMELIST" READ	An illegal character was found in namelist-directed input.	Be sure input data conforms to the syntax rules for namelist-directed input, or remove illegal character from data.
977	ILLEGAL SUB-SCRIPT OR SUB-STRING IN"NAMELIST" READ	An invalid subscript or substring specifier was found in namelist-directed input. Possible causes include bad syntax, subscript/substring component out-of- bounds, wrong number of subscripts and substring on non-character variable.	Check input data for syntax errors. Be sure subscript/substring specifiers are correct for data type. Specify only array elements within the bounds of the array being read.
978	TOO MANY VALUES IN "NAMELIST" READ	Too many input values were found during a namelist-directed READ. This message will be generated by attempts to fill variables beyond their memory limits.	Supply only as many values as the length of the array.
979	VARIABLE NOT IN NAMELIST GROUP	A variable name was encountered in the input stream that was not declared as part of the current namelist group.	Read only the variables in this namelist.
980	NAMELIST I/O ATTEMPTED ON UNFORMATTED FILE	An illegal namelist-directed I/O operation was attempted on an unformatted (binary) file.	Specify FORM='FORMATTED' in OPEN statement, or use namelist-directed I/O only on formatted files.
1010	OPEN WITH ILLEGAL PAD SPECIFIER ATTEMPTED	An attempt was made to open a file with an illegal value specified for the PAD= specifier.	Specify either PAD='YES' or PAD='NO'.
1011	OPEN WITH ILLEGAL POSITION SPECIFIER ATTEMPTED	An attempt was made to open a file with an illegal value specified for the POSITION= specifier.	Specify POSITION='ASIS', POSITION='REWIND' or POSITION='APPEND'.

Error No.	Error Message	Description	Action
1012	OPEN WITH ILLEGAL DELIM SPECIFIER ATTEMPTED	An attempt was made to open a file with an illegal value specified for the DELIM= specifier.	Specify DELIM= 'APOSTROPHE', DELIM='QUOTE' or DELIM='NONE'.
1013	OPEN WITH ILLEGAL ACTION SPECIFIER ATTEMPTED	An attempt was made to open a file with an illegal value specified for the ACTION= specifier.	Specify ACTION='READ', ACTION='WRITE' or ACTION='READWRITE'.

Abbreviations & Acronyms

A2P	Algorithm-to-Product Conversion
ALT	Altimeter or Altimetry, also designation for the EOS-Altimeter spacecraft series
ANCxx	GLAS Ancillary Data Files
APID	GLAS Level-0 Data file
ATBD	Algorithm Theoretical Basis Document
ATM	Atmosphere
CCB	Change Control Board
ClearCase	GSAS version tracking software
CR	Change Request
DAAC	Distributed Active Archive Center
DEM	Digital Elevation Model
DFD	Data Flow Diagram
DLT	Digital Linear Tape
EDOS	EOS Data and Operations System
EDS	Expedited Data Set
ELEV	Elevation
EOC	EOS Operating Center
EOS	NASA Earth Observing System Mission Program
EOSDIS	Earth Observing System Data and Information System
GB	Gigabyte
GDS	GLAS Ground Data System
GLAS	Geoscience Laser Altimeter System instrument or investigation
GLAxx	GLAS Science Data Product Files
GLOP	GLAS Level-0 PGE (correctly called GLAS_L0proc)
GPS	Global Positioning System
GSAS	GLAS Science Algorithm Software
GSFC	NASA Goddard Space Flight Center at Greenbelt, Maryland
GSFC/WFF	NASA Goddard Space Flight Center/Wallops Flight Facility at Wallops Island, Virginia

HDF	Hierarchical Data Format
HDF-EOS	EOS-specific Hierarchical Data Format
I-SIPS	Icesat Science Investigator Led Processing System
I/O	Input/Output
ICESAT	Ice, Cloud and Land Elevation Satellite
ID	Identification
IDL	Interactive Data Language
IEEE	Institute for Electronics and Electrical Engineering
ISF	Instrument Support Facility
IST	Instrument Star Tracker
KB	Kilobyte
L0	Level 0
L1A	Level-1A
L1B	Level-1 B
L2	Level-2
LASER	Light Amplification by Stimulated Emission of Radiation
LIDAR	Light Detection and Ranging
LPA	Laser Pointing Array
LRS	Laser Reference System
MB	Megabyte
MET	(context sensitive) Mission Elapsed Time or Meteorological
MOSS	Mission Operations Science Simulation (test)
N/A or NA	Not (/) Applicable
NASA	National Aeronautics and Space Administration
NOAA	National Oceanic and Atmospheric Administration
P2A	Product-to-Algorithm Conversion
PAD	Precision Attitude Determination
PDF	Portable Document Format
PDS	Production Data Set
PGE	Product Generation Executable
POD	Precision Orbit Determination
PR	Problem Report

QA	Quality Assessment
QAP	Quality Assessment Processing
SC	Structure Chart
SCF	Science Computing Facility
SDMP	Science Data Management Plan
SDMS	Scheduling and Data Management System
SDP	Standard Data Products
SRS	Stellar Reference System
SSMP	Science Software Management Plan
SSRF	Science Software Requirements Document
TBD	to be determined, to be done, or to be developed
UNIX	the operating system jointly developed by the AT&T Bell Laboratories and the University of California-Berkeley System Division
UTC	Universal Time Correlation
WF	Waveform

Glossary

aggregate	A collection, assemblage, or grouping of distinct data parts together to make a whole. It is generally used to indicate the grouping of GLAS data items, arrays, elements, and EOS parameters into a data record. For example, the collection of Level 1B EOS Data Parameters gathered to form a one-second Level 1B data record. It could be used to represent groupings of various GLAS data entities such as data items aggregated as an array, data items and arrays aggregated into a GLAS Data Element, GLAS Data Elements aggregated as an EOS Data Parameter, or EOS Data Parameters aggregated into a Data Product record.
array	An ordered arrangement of homogenous data items that may either be synchronous or asynchronous. An array of data items usually implies the ability to access individual data items or members of the array by an index. An array of GLAS data items might represent the three coordinates of a georeference location, a collection of values at a rate, or a collection of values describing an altimeter waveform.
file	A collection of data stored as records and terminated by a physical or logical end-of-file (EOF) marker. The term usually applies to the collection within a storage device or storage media such as a disk file or a tape file. Loosely employed it is used to indicate a collection of GLAS data records without a standard label. For the Level 1A Data Product, the file would constitute the collection of one-second Level 1A data records generated in the SDPS working storage for a single pass.
header	A text and/or binary label or information record, record set, or block, prefacing a data record, record set, or a file. A header usually contains identifying or descriptive information, and may sometimes be embedded within a record rather than attached as a prefix.
item	Specifically, a data item. A discrete, non-decomposable unit of data, usually a single word or value in a data record, or a single value from a data array. The representation of a single GLAS data value within a data array or a GLAS Data Element.
label	The text and/or binary information records, record set, block, header, or headers prefacing a data file or linked to a data file sufficient to form a labeled data product. A standard label may imply a standard data product. A label may consist of a single header as well as multiple headers and markers depending on the defining authority.
Level 0	The level designation applied to an EOS data product that consists of raw instrument data, recorded at the original resolution, in time order, with any duplicate or redundant data packets removed.
Level 1A	The level designation applied to an EOS data product that consists of reconstructed, unprocessed Level 0 instrument data, recorded at the full resolution with time referenced data records, in time order. The data are annotated with ancillary information including radiometric and geometric calibration coefficients, and georeferencing parameter data (i.e., ephemeris data). The included, computed coefficients and parameter data have not however been applied to correct the Level 0 instrument data contents.

Level 1B	The level designation applied to an EOS data product that consists of Level 1A data that have been radiometrically corrected, processed from raw data into sensor data units, and have been geolocated according to applied georeferencing data.
Level 2	The level designation applied to an EOS data product that consists of derived geophysical data values, recorded at the same resolution, time order, and georeference location as the Level 1A or Level 1B data.
Level 3	The level designation applied to an EOS data product that consists of geophysical data values derived from Level 1 or Level 2 data, recorded at a temporally or spatially resampled resolution.
Level 4	The level designation applied to an EOS data product that consists of data from modeled output or resultant analysis of lower level data that are not directly derived by the GLAS instrument and supplemental sensors.
metadata	The textual information supplied as supplemental, descriptive information to a data product. It may consist of fixed or variable length records of ASCII data describing files, records, parameters, elements, items, formats, etc., that may serve as catalog, data base, keyword/value, header, or label data. This data may be parsable and searchable by some tool or utility program.
orbit	The passage of time and spacecraft travel signifying a complete journey around a celestial or terrestrial body. For GLAS and the EOS ALT-L spacecraft each orbit starts at the time when the spacecraft is on the equator traveling toward the North Pole, continues through the equator crossing as the spacecraft ground track moves toward the South Pole, and terminates when the spacecraft has reached the equator moving northward from the South Polar region.
model	A graphical representation of a system.
module	A collection of program statements with four basic attributes: input and output, function, mechanics and internal data.
parameter	Specifically, an EOS Data Parameter. This is a defining, controlling, or constraining data unit associated with a EOS science community approved algorithm. It is identified by an EOS Parameter Number and Parameter Name. An EOS Data Parameter within the GLAS Data Product is composed of one or more GLAS Data Elements
pass	A sub-segment of an orbit, it may consist of the ascending or descending portion of an orbit (e.g., a descending pass would consist of the ground track segment beginning with the northernmost point of travel through the following southernmost point of travel), or the segment above or below the equator; for GLAS the pass is identified as either the northern or southern hemisphere portion of the ground track on any orbit
PDL	Program Design Language (Pseudocode). A language tool used for module programming and specification. It is at a higher level than any existing compilable language.
process	An activity on a dataflow diagram that transforms input data flow(s) into output data flow(s).

product	Specifically, the Data Product or the EOS Data Product. This is implicitly the labeled data product or the data product as produced by software on the SDPS or SCF. A GLAS data product refers to the data file or record collection either prefaced with a product label or standard formatted data label or linked to a product label or standard formatted data label file. Loosely used, it may indicate a single pass file aggregation, or the entire set of product files contained in a data repository.
program	The smallest set of computer instructions that can be executed as a stand-alone unit
record	A specific organization or aggregate of data items. It represents the collection of EOS Data Parameters within a given time interval, such as a one-second data record. It is the first level decomposition of a product file.
Scenario	A single execution path for a process.
Standard Data Product	Specifically, a GLAS Standard Data Product. It represents an EOS ALT-L/GLAS Data Product produced on the EOSDIS SDPS for GLAS data product generation or within the GLAS Science Computing Facility using EOS science community approved algorithms. It is routinely produced and is intended to be archived in the EOSDIS data repository for EOS user community-wide access and retrieval.
State Transition Diagram	Graphical representation of one or more scenarios.
Stub	(alias dummy module) a primitive implementation of a subordinate module, which is normally used in the top-down testing of superordinate (higher) modules.
Structured Chart	A graphical tool for depicting the partitioning of a system into modules, the hierarchy and organization of those modules, and the communication interfaces between the modules.
Structured Design	The development of a blueprint of a computer system solution to a problem, having the same components and interrelationships amount the components as the original problem has.
Subroutine	A program that is called by another program
variable	Usually a reference in a computer program to a storage location, i.e., a place to contain or hold the value of a data item.