193-801-SD4-001

EOSDIS Core System Project

PGS Toolkit Requirements Specification for the ECS Project

Draft Version 3.1

September 10, 1993

Hughes Applied Information Systems, Inc.
Landover, Maryland

PGS Toolkit Requirements Specification for the ECS Project

Draft

September 10, 1993

Prepared Under Contract NAS5-60000

APPROVED BY

_____

Marshall A. Caplan, Project Manager — Date
EOSDIS Core System Project

Hughes Applied Information Systems, Inc.
Landover, Maryland

This page intentionally left blank.

Preface

This draft PGS Requirements Specification will be a living document until a final version is delivered to NASA and the ECS scientific community in October, 1993. The purpose of this draft is to solicit comments from the community. The initial Toolkit requirements were derived from a NASA study (The PGS Toolkit Study Report, V1.9, NASA-ESDIS, received May 7, 1993) and from the Functional and Performance Requirements Specification for the EOSDIS Core System (NAS5-60000). A preliminary draft of the present document was delivered for review to the NASA ESDIS Project office and investigator teams in late June, 1993. Refinements were made to the preliminary draft based upon reviews and consultations with the NASA ECS Project, the AIRS, ASTER, CERES, LIS, MISR, MODIS, and MOPITT science software development teams. Additional refinement resulted from PGS operational design clarification. It is expected that further requirements and refinements will come from continued consultation with additional EOS instrument teams and investigators. Toolkit design and implementation plans must be made in conjunction with the scientific community as well as system designers from the Hughes Team.

This page intentionally left blank.

Change Information Page

List of Effective Pages	
Page Number	**Issue**
Title	Draft
iii through x	Draft
1-1 through 1-4	Draft
2-1 and 2-2	Draft
3-1 through 3-11	Draft
4-1 through 4-3	Draft
5-1 through 5-6	Draft
6-1 through 6-11	Draft
7-1 through 7-30	Draft
8-1	Draft
A-1 through A-6	Draft
AB-1 through AB-3	Draft
GL-1 through GL-2	Draft

Document History			
Document Number	**Status/Issue**	**Publication Date**	**CCR Number**
193-801-SD4-001	Draft	September 1993	

This page intentionally left blank.

Contents

Preface

1. Introduction

1.1 Identification 1-1

1.2 Scope 1-1

1.3 Purpose and Objectives 1-1

1.4 Status and Schedule 1-2

1.5 Document Organization 1-2

2. Related Documentation

2.1 Parent Documents 2-1

2.2 Applicable Documents 2-1

3. ECS Overview

3.1 Introduction 3-1

3.2 The Science Data Processing Segment 3-1

3.2.1 PGS — Product Generation System 3-1

3.2.2 DADS — Data Archive and Distribution System 3-2

3.2.3 IMS — Information Management System 3-2

3.3 The Product Generation System Environment 3-2

3.3.1 PGS Environment Overview 3-2

3.3.2 PGS Interfaces 3-3

3.3.3 Processing 'String' 3-4

3.3.4 Product Generation Executive 3-5

3.3.5 PGS Scheduling 3-7

3.3.6 PGE Execution 3-9

3.3.7 Science Software Development 3-10

3.4 Science Processing 3-10

4. PGS and SCF Toolkit General Objectives

4.1 Interfaces 4-1

4.2 Capacity and Performance 4-1

4.3 Portability and Adaptability 4-2

4.4 Maintainability and System Evolution 4-2

4.5 Security 4-2

4.6 Data Access 4-2

4.7 Implementation Plan 4-2

4.8 Operational Capabilities 4-3

4.9 Standards 4-3

4.10 Quality Assurance of Toolkit Software 4-3

4.11 Technical Support and Documentation 4-3

5. SCF Toolkit and Development Environment

5.1 Purpose 5-1

5.2 Requirements 5-1

5.2.1 General 5-1

5.2.2 Tool Categories 5-2

5.2.3 Code and Hardware 5-3

5.2.4 Data Access 5-3

5.2.5 Support and Documentation 5-3

5.3 SCF and PGS Environmental Comparison 5-4

5.3.1 Overview 5-4

5.3.2 Planner (formerly Scheduler) 5-4

5.3.3 Execution Monitor 5-4

5.3.4 Toolkit functions 5-4

5.3.5 Other Development Tools 5-5

5.3.6 Science Software Delivery, Integration and Test 5-5

6. PGS Toolkit Requirements

6.1 Parental (Level 3) Requirements 6-1

6.2 PGS Toolkit Requirements - System Mandatory 6-2

6.2.1 File I/O Tools 6-3

6.2.2 Error/Status Reporting 6-5
6.2.3 Process Control 6-5
6.2.4 Memory Management 6-5
6.2.5 Ancillary Data Access and Manipulation 6-5
6.2.6 Time and Date Conversion Tools 6-8
6.3 PGS Toolkit Requirements - Science 6-8
6.3.1 Planetary Body and Spacecraft Position Access Tools - Mandatory 6-8
6.3.2 Math & Modeling Support Tools - Optional 6-10
6.3.3 Constants and Unit Conversions - Optional 6-10
6.3.4 Graphics Support Tools - Optional 6-11

7. PGS Toolkit Specification

7.1 Introduction 7-1
7.2 PGS Toolkit Tools - System Mandatory 7-1
7.2.1 File I/O Tools 7-1
7.2.2 Error/Status Reporting Tool Requirements 7-10
7.2.3 Process Control Tools 7-14
7.2.4 Memory Management Tools 7-15
7.2.5 Ancillary Data Access and Manipulation Tools 7-27
7.2.6 Time and Date Conversion Tools 7-20
7.3 PGS Toolkit Tools - Science 7-23
7.3.1 Planetary Body and Spacecraft Position Access Tools - Mandatory 7-23
7.3.2 Math & Modeling Support Tools - Optional 7-28
7.3.3 Constants and Unit Conversions - Optional 7-29
7.3.4 Graphics Support Tools - Optional 7-30

8. Proposed PGS Toolkit Delivery Priorities

Figures

3-1 PGS Interfaces 3-4
3-2 SDPS Architecture Showing the Production String Concept 3-5
3-3 Product Generation Executable Concept 3-6
3-4 Scheduling Scenario 3-8
3-5 Product Processing 3-11

Tables

5-1 A Comparison of SCF and PGS Functionality 5-5

Appendix A. Requirements Traceability Matrix

Appendix B. POSIX System Calls Usage Policy

Appendix C. Open Issues

Appendix D. Disposition of Unused Requirements

Abbreviations and Acronyms

Glossary

1. Introduction

1.1 Identification

This Draft ECS PGS Toolkit Requirements Specification is a deliverable document under the Earth Observing System Data and Information System (EOSDIS) Core System (ECS), Contract (NAS5-60000). It follows a preliminary draft version which was provided for review in late June, 1993.

1.2 Scope

The Draft ECS PGS Toolkit Requirements Specification defines the requirements for the ECS Product Generation System (PGS) science production software and development Toolkit. This document provides an overview of the DAAC architecture environment in which the science production software will run. A detailed list of requirements is provided for tools which will be used to integrate and maintain science software into the PGS. Requirements for additional tools, which will be used solely in the science software development environment, are provided as well. A preliminary description of tools to which requirements have been allocated is included. Finally, a list of priorities for software deliveries by tool category is provided.

1.3 Purpose and Objectives

This document is a draft requirements specification of the ECS PGS Toolkit. The ECS science software developer will use the Toolkit to access the PGS environment and services. The required use of the Toolkit can be divided into three classes:

System Mandatory — In the production environment, external calls from the science software, for system and resource accesses, file I/O requests, error message transaction, metadata formatting and geographic information database requests will be made through Toolkit calls. The use of these tools will be enforced through automatic checks at integration time at the DAACs.

Science Mandatory — These are the parts of the Toolkit that are necessary to ensure that the products can be used in subsequent processing and in correlative studies involving more than one type of EOS product. These type of calls include geolocation, time conversions, and physical constants. The use of these tools is more difficult to verify using automated code checkers, but will be assessed at integration time using a combination of automated checks (for inclusion and exclusion) and manual code reviews.

Science Optional — Other useful services such as scientific and math library calls, will be provided by the Toolkit. The use of these services is optional, but is encouraged. Algorithm developers who use alternative solutions will be required to deliver the source code for the replacement services as part of the algorithm delivery.

The PGS Toolkit will serve to insulate science software from the PGS architecture, and to provide a development environment which emulates critical DAAC PGS functions. The Toolkit will help ensure code portability as the algorithm is ported from development hardware, through the DAAC system, and through potential hardware changes as the ECS matures. To do so effectively, the Toolkit will provide for limited access and control to system level resources, including processes, memory, and I/O capabilities. Where control of such resources is necessary (e.g. dynamic memory allocation), the Toolkit will provide a set of routines through which the application must obtain those services. This partitioning and layering of operating system services allows the Toolkit to work on behalf of

the PGS Scheduler in allocating, de-allocating, and making use of system-wide shared resources. Appendix B describes this use policy for the POSIX system calls, as defined in IEEE Std 1003.1 and IEEE Std 1003.9.

The Toolkit will also serve to minimize code development by providing common functionality required across the ECS community.

The requirements for much of the Toolkit software comes from contacts with the scientific community, through science operations scenarios and requirements reviews. Our objective is to provide the science community with a list of tools which will aid in the science software development process.

The PGS Toolkit will be delivered in stages to the developers. The priority of deliveries is an important part of the requirements specification and will be described in detail in Section 8 below. The software contained in these deliveries represent a baseline which will satisfy the identified functional requirements. The baseline will be modified as additional functional requirements arise during the ECS program lifetime.

The principal goal of this document is to present a compilation of Toolkit requirements. This compilation has resulted from an investigation of science software developers needs, their development environments, DAAC system design requirements, and requirements imposed by instrument development and flight schedules.

The requirements on the Toolkit which are listed in this document are traced to higher level parent requirements. In the case where no trace exists, disposition of that particular need will be accomplished through normal program "recommended requirements" gathering.

1.4 Status and Schedule

A preliminary draft of this requirements document was provided to the community shortly after the Data Processing Focus Team (DPFT) meeting on June 23, 1993. The final version will be delivered to NASA and to the ECS community in October, 1993. The development of this document was initiated upon receipt of the NASA PGS Toolkit Study Report, V1.9, on May 7, 1993. Successive versions contain revisions and refinements based on inputs and feedback from science software development teams, associated with instruments to be flown on the EOS AM, EOS PM and TRMM spacecrafts. The final requirements specification document will be followed shortly by a software design document and an implementation and test plan for software deliveries to the DAACs.

1.5 Document Organization

The document is organized as follows:

Section 1	Introduction - Presents the scope and purpose of this document.
Section 2	Related Documentation - Provides a bibliography of reference documents for the PGS Toolkits organized by parent and applicable documents.
Section 3	ECS Overview - Gives a description of the DAAC architecture in which the science product generation software will reside.
Section 4	PGS and SCF Toolkit General Objectives - Provides a list of general requirements on security, maintainability, system evolution, etc.
Section 5	SCF Toolkit and Development Environment - Provides a list of requirements for software development tools.
Section 6	PGS Toolkit Requirements - Provides a detailed list of production environment tools along with their functions and description.
Section 7	PGS Toolkit Specification - A preliminary list of tools allocated to the requirements in Sections 5 and 6.

Section 8	Priorities of Software Deliveries - Provides a prioritized list of software deliveries.
Appendix A	Requirements Traceability Matrix - A trace of requirements in Section 6 to Level 3 Functional Requirements.
Appendix B	POSIX System Calls Usage Policy - Provides a list of prohibited and allowed operating system calls.
Appendix C	Open Issues - Provides a list of expressed needs which are under consideration as Toolkit requirements.
Appendix D	Disposition of Unused Requirements - Explains why certain potential requirements received from the science software instrument teams were not used as PGS Toolkit requirements.
Glossary	
Acronyms	

This page intentionally left blank.

2. Related Documentation

2.1 Parent Documents

The following documents are the parents from which this document's scope and content derive:

The PGS Toolkit Study Report, Version 1.9a, GSFC-ESDIS 5/6/93

Proposal for EOSDIS Core System: Technical Proposal, 9/3/91, Hughes Team

Functional and Performance Requirements Specification for the Earth Observing Data and Information System (EOSDIS) Core System, NAS5-60000 Attachment B 2/16/93

EOSDIS Core System Statement of Work, 16 February 1993, NASA/GSFC

EOSDIS Core System Contract Data Requirements Document, GSFC 2/16/93

NASA Software Documentation Standard Software Engineering Program, NASA-STD-2100-91 7/29/91

ECS Project Management Plan,101/MG1 5/93

2.2 Applicable Documents

The following documents are directly applicable to this requirements specification:

UARS Lessons Learned for EOS: Report 1—Design and Implementation, CSC document for NASA contract NAS5-31500

NCSA HDF Calling Interfaces and Utilities, Version 3.2, March 1993

ECS System Implementation Plan, 301/DV1 5/93

Science Users Guide and Operations Handbook, Volume 4: Algorithm Development and Integration. HAIS August 1993. [DID 205/SE1-006]

IEEE Std 1003.1:POSIX Part 1: System Application Program Interface (API)[CLanguage]

IEEE Std 1003.9:POSIX FORTRAN77 Language Interfaces, Part 1: Binding for System Application Program Interface [API]

This page intentionally left blank.

3. ECS Overview

3.1 Introduction

The Earth Observing System Data and Information Core System (ECS) will be distributing a collection of institutional assets, facilities, and core system functions that work together to support EOS and Mission to Planet Earth (MTPE). ECS is the overall integrating infrastructure for operations as well as a provider of data and information services for EOS mission support, science data processing, communications, and system management. A feature of ECS is its ability to interoperate with other existing and developing systems to take maximum advantage of their capabilities without the need to assimilate these other systems into ECS. ECS is viewed as the evolutionary base for accelerating the pace and quality of Earth science research.

In this Section, we describe the segment of the ECS which lies within the scope of this document, the Science Data Processing Segment. Particular attention is given to a description of the environment of the Product Generation System, a component of the SDPS.

3.2 The Science Data Processing Segment

The goals of the Science Data Processing Segment (**SDPS**) are to produce, archive, and maintain long term global science research products, to provide the means to search its holdings, and to deliver ordered products.

The SDPS consists of a distributed Information Management System (**IMS**), and a set of Distributed Active Archive Centers (**DAACs**), each housing a Product Generation System (**PGS**) and a Data Archive and Distribution System (**DADS**). This fundamental architecture is consistent across the DAACs, although individual DAACs may be configured with different hardware platforms to meet site specific processing needs.

3.2.1 PGS — Product Generation System

The Product Generation System (PGS) employs scientist-developed algorithms to create data "products", datasets defined by the instrument teams as having significance in global climate change research. PGS processing is driven both by data ingested for product generation (a "push" on the system) and by user requests for data retrieval (a "pull" on the system). Separate user request and product generation networks greatly limit query response and production performance interference. Multimedia handling is required to support both input and output to users and interoperability with other systems. The ingest and distribution subsystems economically share multimedia peripherals and operations staff. Each PGS site produces its own unique set of "products", datasets defined by the instrument teams as having significance in the scientific community. Within a PGS, production is accomplished by the scheduling and execution subsystem (SES), which pre-stages data necessary for product algorithm initiation. Output data is archived, and metadata, browse and quality assurance data are produced and routed to appropriate locations as described below. The processing is segmented by product and executed by processing "strings", specific computer hardware configurations that have been arranged to provide adequate machine resources (processing, memory, and I/O) for product generation. A dedicated test/backup string separates algorithm test and integration, prototype product generation, and training activities from operational processing and reprocessing, and provides backup processing to work around component failures and routine maintenance downtime. The string concept is described in more detail later in this section.

3.2.2 DADS — Data Archive and Distribution System

The Data Archive and Distribution System (**DADS**) provides for the ingesting of source instrument data, and the archiving, management, and distribution of data products according to PGS processing needs and users' requests. The DADS provides a service-oriented interface to other system components, managing the large storage hierarchy transparently to facilitate infusion of evolving technologies and to provide a uniform storage model. The DADS provides for prompt data distribution in multiple formats, including direct electronic distribution (via the Internet, for example), and magnetic and optical media.

3.2.3 IMS — Information Management System

The Information Management System (**IMS**) maintains information about the wide range of data products within the ECS, and provides search, browse, and data ordering capabilities to the science community at large through networked terminal- and Windows-based interfaces. The IMS provides access to a wide range of data, including product metadata, algorithm implementation information, relevant scientific journal articles, and project-oriented bulletin boards. A system "data map" ensures that searches will execute successfully across inconsistencies between components of the heterogeneous system (e.g., transparently equates "sensor" and "instrument" between systems). The ECS inventory (metadata) that the IMS manages is partitioned and stored at the various DAACs, employing a partitioned database management system.

3.3 The Product Generation System Environment

In this section, we describe the production environment in which the science software will reside. The interfaces, scheduling system, standard product generation, and other concepts presented will bear a direct relation to requirements for Toolkit software in Section 6 below. Understanding these concepts is vital to constructing a science software development environment at the Science Computing Facilities (SCF). Toolkit requirements for the SCF environment are presented in Section 5.

3.3.1 PGS Environment Overview

The Product Generation System (PGS) is a production oriented computing facility which executes science software to generate Standard Products from instrument and other types of data. Standard Products include Level 1 through

Level 4 data products, browse, and quick-look products. These products are stored in the DADS for general access by scientists and other users of the ECS system (quick-look products are also delivered to the ICC and field campaign users). The PGS produces metadata to facilitate selection of Standard Products from archival holdings and interpretation of the products through the IMS. It generates data quality assessment profiles for each Standard Product in coordination with the scientists. It reprocesses large portions of data sets to provide a consistent, long-term database that meets high standards of scientific quality.

Scientists submit both new and enhanced software to the PGS to produce additional Standard Products, and to enhance existing products in response to changing scientific evaluation results. As part of the life-cycle, some software may be run for a time in a testing mode, producing prototype products which are treated in all other respects by the SDPS as Standard Products.

3.3.2 PGS Interfaces

PGS interfaces are illustrated in Figure 3-1. The primary PGS interface is with its associated DADS. The DADS provides input data files necessary to meet the production schedule, including Level 1 through Level 4 data sets (either residing at the local DADS, or received by the local DADS from remote DADS) and ancillary data. The DADS provides storage for any algorithms and calibration data that the PGS does not maintain on a regular basis. The DADS receives and responds to requests for the staging of stored data. The DADS receives and stores generated Level 1 to Level 4 Standard Products.

The IMS provides the PGS with processing requests from the science community. These include standing orders for Standard Products that will be generated on a regularly-scheduled basis and specific requests for other Standard Products. The PGS reports schedule conflicts to the System Management Center (SMC). Copies of the PGS processing schedules are provided to the IMS to allow users to monitor the status of their requests.

The PGS sends regular status reports and copies of the PGS schedules to the SMC. The SMC delivers operational directives including priority assignments and resolution of schedule conflicts not resolved by consultation with the IMS.

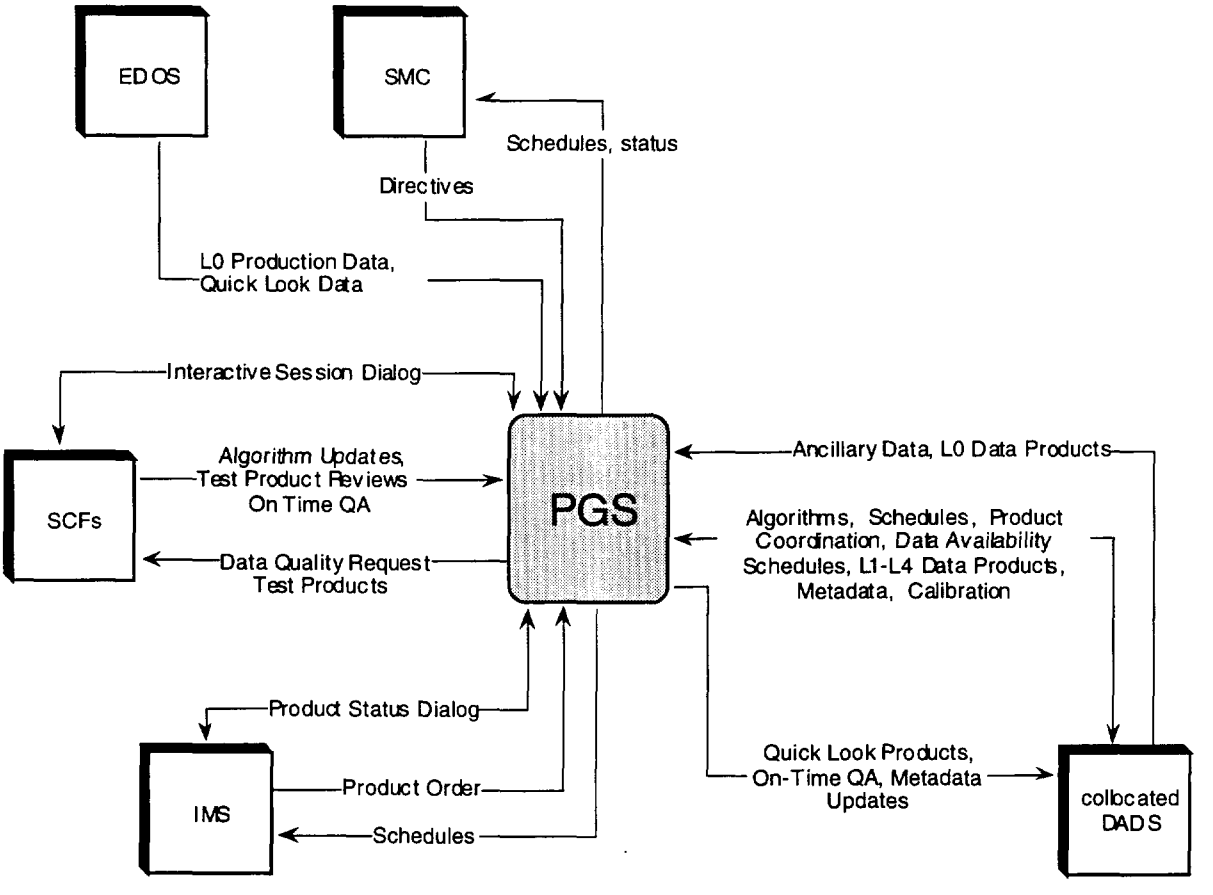
Test products generated by the pre-operational science software are sent to the software developers at the relevant SCF. Reviews of the test products are sent to the PGS. The PGS also sends requests to scientists for product quality assessment and receives the replies.

3.3.3 Processing 'String'

The functional design of the PGS is identical at each DAAC, as illustrated in Figure 3-2. It is within this design that accommodation of the widely varying processing requirements occurs; from simple instrument processing to complex global change models. The physical design utilizes a "string" architecture to support the unique processing requirements of each site. Scheduling is supported by a server that controls the resources of the strings.

In the string architecture (right side of Figure 3-2), hardware is laid out as a series, or "string" of equipment designed to perform processing and reprocessing. Processing is assigned to strings to minimize contention between them and keep each one directed to production of specific products. The production strings are connected to the DADS through a separate high speed network. Each individual string is designed to support the unique processing and reprocessing requirements of the product generation tasks assigned to it. The types of algorithms, the data needs, and the class of machine the algorithm requires are all used to define the physical components of the string. Small sites use fewer strings and/or smaller processors than the larger sites, but they all have the same string architecture. Sites can grow or shrink over time as necessary to accommodate changing product generation requirements. A separate test/backup string is supplied and sized to backup any other string in the DAAC with identical architectural processing capabilities. It is used for algorithm test and integration, for production of prototype products, for backup during critical failures of any other string, and to support training.

Figure 3-1. PGS Interfaces



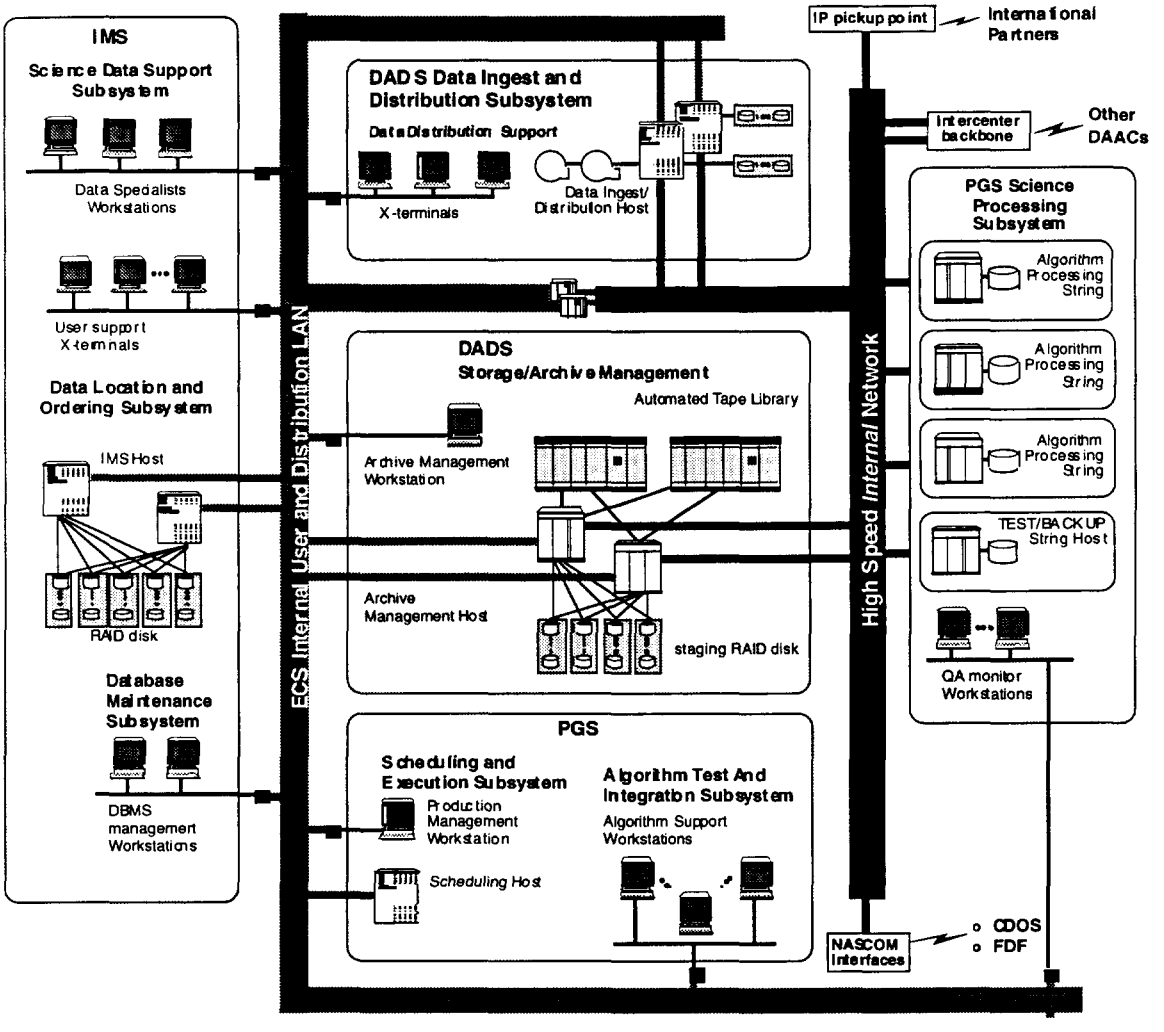


Figure 3-2. SDPS Architecture Showing the Production String Concept

3.3.4 Product Generation Executive

Science software is executed as a Product Generation Executive (PGE). A PGE may consist of one or more compiled binary executables and/or command language scripts. A PGE produces one or more standard or intermediate product files, or completes a processing step on a data set as shown in Figure 3-3. In most cases, the PGE will consist of a script which links together executables to perform the processing. PGEs are the smallest schedulable unit for PGS processing, and are presented to the scheduling subsystem along with a set of inputs, a sequence of processing steps, and a set of outputs. When all of the inputs are available for processing, the set of inputs are staged on local disk storage before any processing is initiated.

Normal processing operations will be performed automatically, but PGEs can be initiated manually to generate products. Individual PGEs can be executed as UNIX shell level commands. For development purposes, temporary production scripts can be built which invoke one or more PGEs.

PGEs can consist of an arbitrary amount of processing. However, there are design guidelines for PGE definition that take system wide scheduling and resource management into consideration. If each PGE can be designed to generate a limited set of products (perhaps only one), several advantages will be realized. Natural checkpoints will occur between PGEs because their products (and temporary products, if any) are sent to the archive.

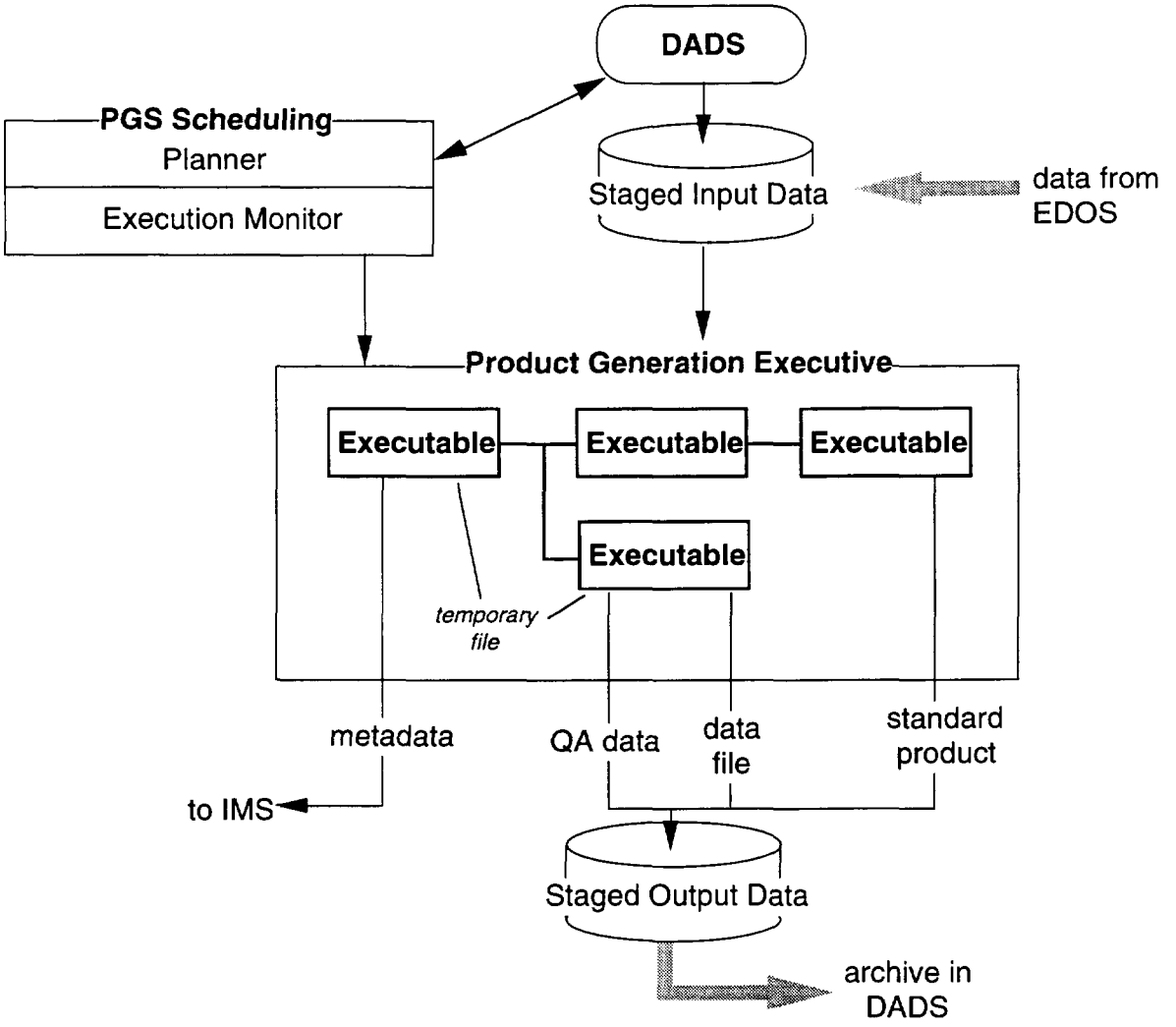


Figure 3-3. Product Generation Executable Concept

If PGEs are established to produce a single product, or an intermediate stage of a product, reprocessing efforts can be simplified. Tasks can be scheduled to perform the minimal amount of processing necessary.

3.3.5 PGS Scheduling

Although product generation is primarily data driven and not schedule driven, some activities within the SDPS (tests, training, hardware maintenance, and some distributions) are performed on a scheduled basis. The PGS Scheduling Execution Subsystem (**SES**) balances the data driven and schedule driven processing requirements by performing the following:

- generate SDPS plans with resolution to the PGE level
- develop processing plans for the next day
- orchestrate product generation tasks
- schedule activities according to established priorities
- monitor deadlines
- flag potential problems such as possible schedule slips

The scheduling process is shown in Figure 3-4.

DAAC activity plans are created using data collection and availability schedules, outstanding product orders, and required support tasks such as hardware maintenance. Activity plans are reviewed by management and operations staff with conflicts reported for resolution.

The scheduling is conducted in two ways:

a. A schedule, or activity plan is generated which reflects the anticipated timeline of tasks to be performed in support of Science Data Processing within a site. This is more of a guideline for activities than an absolute schedule of the time at which activities will be initiated. Some activities within the site (tests, training, hardware maintenance, and some distributions) will be performed on a scheduled/time dependent basis, but product generation is ultimately data, resource and priority driven, not schedule driven.

b. The second scheduling function applies only to product generation, which is initiated whenever all of the required input data for a product generation executive (PGE) is available. The information in the activity plan includes identification and priority of the PGE to be executed, the approximate timeline of the PGE, PGE inputs and outputs, the string to be used, and the resources required within that string. Since the initiation of each PGE was planned based on the predicted availability of input data and the availability of resources, the plan serves primarily as a reference framework to indicate when product generation tasks are falling behind.

The processing policies required to handle anomalous conditions for standard products will be established as part of the algorithm test and integration activity. Science software deliveries from the instrument teams will include the parameters required to submit requests for processing of standard products. The human Production Scheduler will work with the Algorithm Support Team to enter and modify routine processing requests, indicated as **step 1a** in Figure 3-4 above. On demand, or *ad hoc*, processing requests are submitted by authorized scientists through an interface similar to that used to submit data orders. This request is indicated as **step 1b**.

Plans are created based on data collection and availability schedules, product orders, information about available hardware and software resources, and scheduled support activities such as hardware maintenance. Data availability schedules are received from remote archives, EDOS, and other suitable sources as indicated in **step 2**. Plan generation requires information regarding the configuration and status of computers, availability of disk storage, and the current assignment of product generation processing to hardware strings.

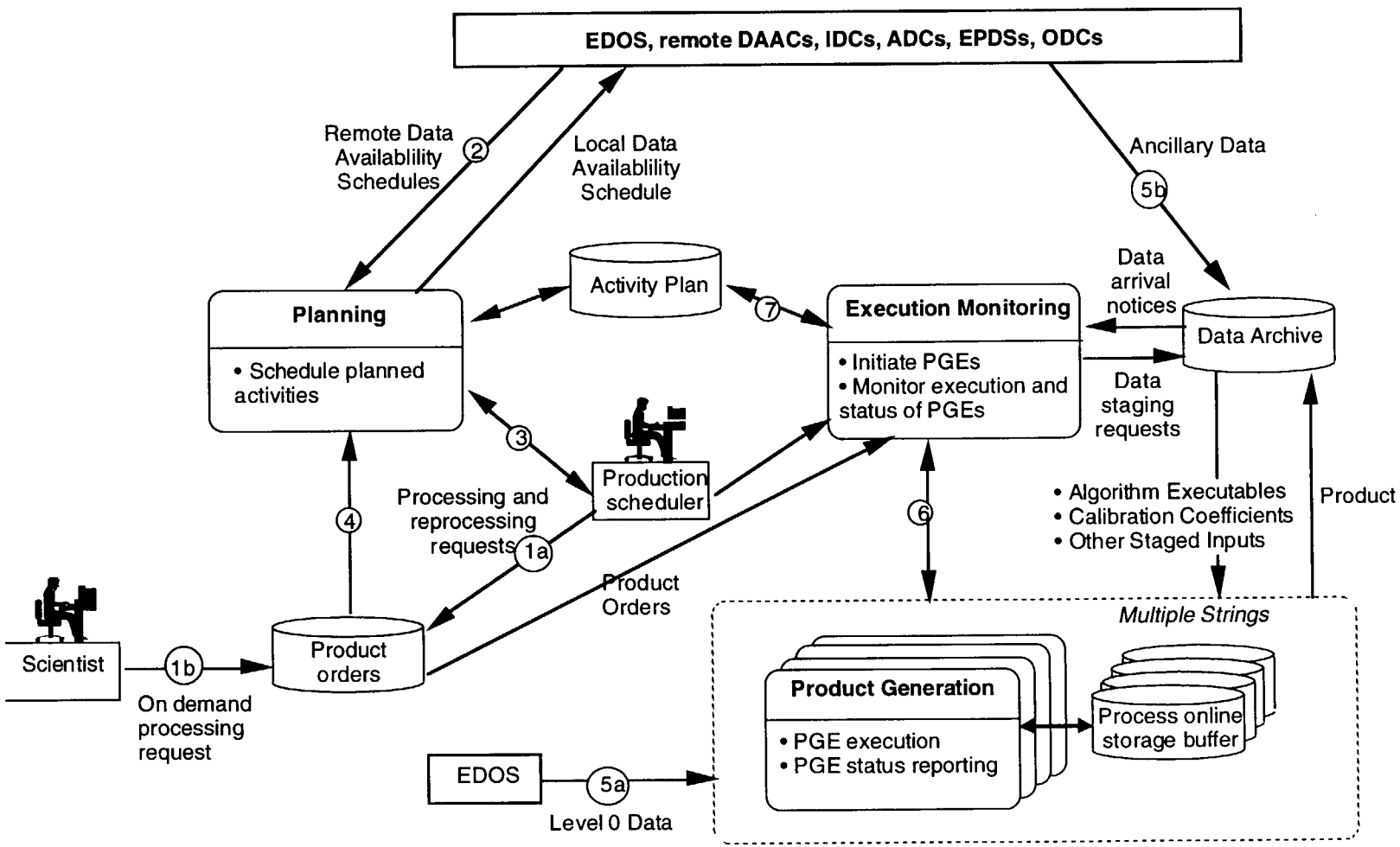


Figure 3-4. Scheduling Scenario

The human Production Scheduler initiates the generation of DAAC activity plans on a daily basis during the normal workweek, **step 3**. These plans cover the same period as the EDOS Data Availability Schedule. Each plan contains an initial firm planning period of approximately 1 day. The rest of the plan is more tentative, and is intended to provide visibility of the extended plan to support reviewers in identifying potential resource conflicts early and in helping to resolve or prevent them.

Product orders are obtained from the Product Orders Database, **step 4**. These can be processing requests, standing orders for standard products, or reprocessing requests. Orders consist of parameters that control product generation and distribution. These parameters include:

- identification of the product to be generated in conjunction with the version of the software to be used.
- the time period during which the designated product should be generated.
- locations for product distribution.

3.3.6 PGE Execution

The execution monitor's event driven processing actually initiates PGEs for product generation. As data arrives in the ECS, it is checked against the PGEs in the activity plan which are waiting for it. EDOS notifies ECS of the availability of Level 0 data. As soon as ECS is ready for it, the Level 0 data is retrieved and staged directly to the string that will use it (**step 5a**). Any other data required to execute the PGE is staged to the string as it becomes available (**step 5b**). This may include results of metadata searches, metadata itself, or optional data. Once all of the necessary data is staged to the string, processing assignment occurs. The PGE is put on a queue for its target string. As each PGE terminates and releases computer resources (CPU, RAM, virtual memory, etc.), that machine's available resources are evaluated against those required by the next PGE on the queue. As soon as adequate resources are available, execution of the PGE is initiated (**step 6**).

PGE initiations and completions are recorded and monitored in real time against the activity plan (**step 7**). For each instance of a PGE that is to be executed, two time thresholds are calculated. The first threshold is the latest time, based on its anticipated processing time, after which PGE initiation *might* result in a missed processing deadline. The second threshold is the latest time that the PGE can be initiated and still be expected to complete by its processing deadline. If a task initiation is delayed beyond the first threshold , the QA/Production Monitor is informed of the *potential* for a missed deadline and as to the reason for the delay (lack of data, processing load, etc.). If the second, more definite, initiation threshold is missed, it will be assumed that the task cannot be accomplished through normal operations. The QA/Production Monitor is informed that the task cannot be completed on schedule. The Production Scheduler or operations staff will then determine the appropriate action to take.

Each string includes sufficient processing bandwidth to handle excessive loads and thus minimize the impact of delays. For example, if EDOS misses a TDRSS contact, the L0 data for that pass will be delayed. On the next pass, two orbits may be down-linked. During the time that processing is delayed due to late delivery of L0 data from EDOS, the reprocessing tasks use the excess capacity to "get ahead" on their workload. When the two passes of L0 Data arrive, the L1 processing uses the additional capacity that is now available due to reduced reprocessing needs, and standard processing makes up the lost time to get back on schedule.

3.3.7 Science Software Development

The development of science product generation software at Science Computing Facilities (SCF) and procedures for integration and testing of this software are given in the ECS Science Users Guide and Operations Handbook (DID 205/SEI-006). It is expected that prior to availability of an operational scheduling execution subsystem at the PGS, an operational IMS, and other DAAC services, some emulation of this functionality will have to be performed to support algorithm development and testing. The PGS Toolkit will be supplemented by software development tools to supply this needed functionality. The Toolkit calling sequences and interfaces will remain the same in both SCF and PGS environments.

In the SCF, the execution of multiple PGE's will be simulated by a production script. The scripting language will contain constructs for defining conditional logic, i.e., if-then-else loops. It will be able to run multiple executables, receive parameters, pass parameters, receive status values from executables, and perform other additional functions necessary for testing software. Various Unix shells and Perl are being considered for use in developing production scripts.

Given this scripting language to construct production scripts, the developer will be able to test science software and Toolkit interfaces at the SCF. Data can be staged manually and multiple PGE's can be strung together to test end-to-end flow. Implementation details of interfaces will be hidden by the Toolkit.

The need for a dynamic scheduling tool that emulates PGS scheduling was expressed by various science teams. Such a tool would support algorithm development modeling and would provide schedule visibility for Q/A operations planning (see Section 7).

The functionality of the PGS Toolkit in the development environment is further discussed in Section 5.

3.4 Science Processing

Having described the basic design behind the Product Generation System, the next step is to outline how this design is used in the generation of a product. A summary of the process is illustrated in Figure 3-5.

Availability messages are received in the ECS for L0 data, L1-L3 products and ancillary data from a number of sources, and are processed automatically. ECS responds to the announcement of data availability by requesting that the data be sent. After each data file is received, the contents are validated against the header. Quick-look data files are immediately registered with the product orders database. For other data files, such as Level 0 and ancillary data, the format is checked and metadata files are produced.

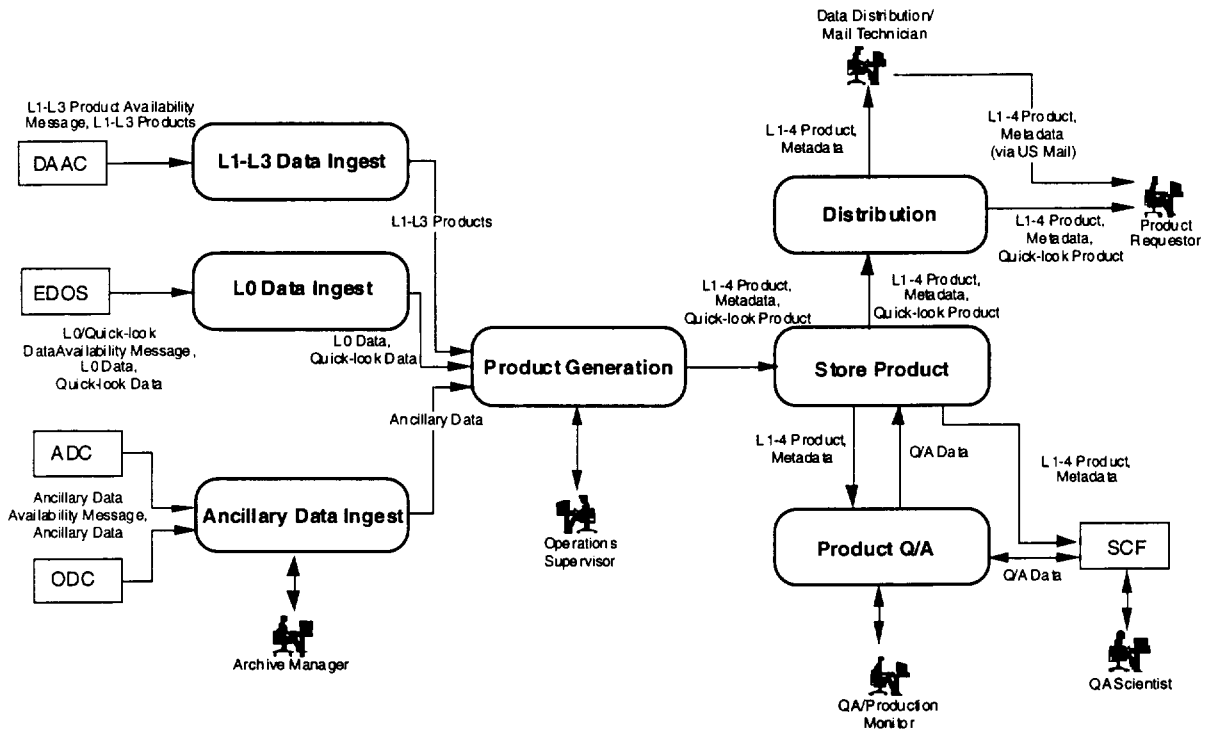


Figure 3-5. Product Processing

Once the incoming Level 0, product and/or quick-look data is received, validated, and registered with the database, the execution monitor is automatically informed. The execution monitor verifies that all the input data required by the PGE is present at the proper location. Then the execution monitor runs the PGE to generate that particular product if the resources are available.

In early releases of the ECS, prior to the delivery of the execution monitor, product generation is initiated manually by the human Production Scheduler, after verifying that all the input data required by the PGE is present at the proper location.

The generation of standard products includes the creation of accompanying metadata during product execution. The quick-look product is distributed to the PI as expeditiously as possible.

The next step in product generation is the routine, automated product quality assurance performed in the PGS. This consists of running a quality assurance executable, provided by the algorithm developer, against the product data. That quality assurance executable performs QA and updates the metadata with a quality assurance code. Following automatic QA, manual quality assurance may be triggered by the QA/Production Monitor if required. This includes a review and possible overriding of the automated QA results and accompanying update of the quality code in the metadata.

If the QA/Production Monitor has noticed a problem, the science team is alerted. The product metadata file is marked to indicate that QA at the SCF is in progress, and the science team is informed that the product file and its accompanying metadata are ready for QA. The science teams then perform QA at their installations; a code flag reflecting the results of their QA analysis is then sent back to the ECS. ECS then automatically updates the product metadata to reflect the QA code, and the product data and its accompanying metadata are then made available to the science community at large and for archiving. If completion of the QA at the SCF is delayed beyond the time frame allotted, ECS automatically marks the product metadata to reflect "product not QA'd" by the scientist and proceeds to make that data available to users and to archive that product data and its accompanying metadata.

4. PGS and SCF Toolkit General Objectives

This Section lists general objectives for the PGS (and SCF development environment) Toolkit. The PGS Toolkit will contain the full functionality required for integration into the production environment and execution of science software in the production environment. A version of the PGS Toolkit (hereafter called the SCF Toolkit) will be delivered to developers. The SCF Toolkit will contain emulation of certain PGS functions and services which are not included in the SCF. Differences between these two versions of the PGS Toolkit are delineated in Section 5. Section 6 lists PGS Toolkit requirements.

Included in this Section are requirements for the interfaces, capacity and performance, portability and adaptability, maintainability, system evolution, security, data access, implementation, standards, technical support services, quality assurance, and documentation.

4.1 Interfaces

4.1.1

The interface services that are provided by the PGS and the SCF Toolkits shall include the following capabilities:

a. delivery of science software, and associated test data and documentation

b. notification of product generation status

c. transmission of quality assurance data sets

d. notification of data quality assurance status

e. science algorithm test products

4.1.2

The SCF Toolkit shall provide interfaces to the SCF that shall perform the following PGS Toolkit functions:

a. initiation, control, and termination of science software execution

b. data management

c. notification to system operators and science users

4.2 Capacity and Performance

4.2.1

The PGS Toolkit shall demonstrate its capacity to handle operational load and to make use of computing resources in an efficient way.

4.2.2

The PGS Toolkit shall provide reports on the computing resources that are used for each version of the science software.

4.3 Portability and Adaptability

4.3.1

The PGS Toolkit shall demonstrate its portability and adaptability by producing the same results in each DAAC's operational environment.

4.3.2

The SCF Toolkit shall exhibit its portability and adaptability by producing the same results on each of the approved computing platforms. Refer to section 5.2.3 Code and Hardware for a list of platforms.

Note: See Section 6.2 for requirements regarding computing languages and environment standards.

4.4 Maintainability and System Evolution

4.4.1

The PGS and SCF Toolkits shall be designed such that they shall demonstrate their maintainability over the system life time when errors are encountered, external environments have changed, or functional enhancements are required.

4.4.2

The PGS and SCF Toolkits shall evolve with enhancements in the science software, and upgrades in the PGS and SCF environments.

4.5 Security

4.5.1

The PGS Toolkit shall adhere to Communications Segment (CSMS) standards for system security.

4.5.2

The SCF Toolkit shall provide the required CSMS standard security needed to operate with the SDPS for science operations.

4.6 Data Access

4.6.1

The PGS Toolkit shall provide access to data sets which are necessary for the generation of science data products.

4.6.2

The SCF Toolkit shall provide access to data sets which are necessary for the development and integration of the science software, and science operations support activities.

4.7 Implementation Plan

4.7.1

The PGS team shall provide a PGS Toolkit implementation plan to delineate capabilities and associated schedule for each version of the DAAC. An implementation plan for the SCF Toolkit shall identify versions of the SCF Toolkit,

and capabilities and the associated schedule for each version. This implementation plan will be provided within the first Toolkit delivery.

4.7.2

The PGS and SCF Toolkits shall incorporate COTS and/or public domain software wherever appropriate.

4.8 Operational Capabilities

4.8.1

The SCF Toolkit shall provide capabilities to configure the Toolkit, control its operation, and support development, integration and operation support activities of the science software.

4.9 Standards

4.9.1

The PGS and SCF Toolkits shall adhere to the ESDIS science software application level standards, related data structures and formats, as well as ECS standards.

4.10 Quality Assurance of Toolkit Software

4.10.1

The process of the development and testing of the quality of the PGS Toolkit shall be consistent with the procedures established at each DAAC and by the ESDIS Project.

4.11 Technical Support and Documentation

4.11.1

Technical support shall be provided to install, configure, and operate the SCF Toolkit, including diagnostic and corrective action services, and maintenance of a log of SCF Toolkit problem reports.

4.11.2

For each version of the SCF Toolkit, documentation and training material shall be provided.

4.11.3

The SCF Toolkit documentation shall include descriptions of Toolkit components , set-up and initialization of the Toolkit, valid configuration options, Toolkit operation, accuracy and precision of Toolkit functions, and associated SCF support.

4.11.4

The SCF Toolkit training document for each version of the SCF Toolkit shall include set-up, initialization, and operation of the Toolkit.

5. SCF Toolkit and Development Environment

5.1 Purpose

The purpose of the SCF development environment and toolkit is (1) to provide development toolkit functions that emulate the production toolkit functions, (2) to provide a development environment that emulates the production environment to the extent possible, (3) make both functions and environment easy to use, and (4) most importantly, allow for a smooth transition of science software from the SCF to the PGS, during the integration and test phase. All PGS toolkit routines will be provided to EOS science software developers in SCF versions.

It is essential to understand the concepts that distinguish the SCF development environment from the PGS production environment. While the science software and interface to the PGS Toolkit are preserved in both environments, there may be slightly different implementations and behavior in the toolkit functions and peripheral components (e.g. shell level development and testing tools). This section highlights the functionality of the PGS Toolkit unique to the SCF. Section 6 of this document lists requirements for specific toolkit functionality which will be used in the production and development environments.

Note that the requirements listed below are not associated with specific Toolkit routines, unlike those which appear in sections 6 and 7. Hence there are no tool descriptions associated with these requirements.

5.2 Requirements

5.2.1 General

Requirement:

PGSTK-0020 Calling sequences of SCF Toolkit functions and PGS Toolkit functions shall be identical.

Notes:

In order for toolkit routines to be integrated into both the development and the production versions of the science software, the actual physical calling sequence in the code must be exactly the same in both versions. Note that there may be differences in the usage of the parameters passed to the functions; this is discussed elsewhere in this section.

Requirement:

PGSTK-0010 The interfaces between the SCF Toolkit functions and the science software shall either be identical to the interfaces between the PGS Toolkit functions and the science software, or emulated.

Notes:

Interfaces to the Toolkit include such items as the scheduling system, the communications system, and the computer operating system. Interfaces such as those to the communications system will need to be emulated. Interfaces such as those with the computer operating system will be identical.

Requirement:

PGSTK-0040 Logical file paths referenced by SCF Toolkit functions and PGS Toolkit functions shall be identical, i.e. all file references shall be by logical file names.

Notes:

In order to make transition between SCF development environment and PGS production environment easier, file names will be represented by logicals, i.e. by Unix environment variables. (The means of translating the logical values may differ between the two environments.) This requirement also makes it easier to assure that files shared among different executables in the SCF environment are accessed in a consistent manner.

5.2.2 Tool Categories

Requirements:

PGSTK-0050 The SCF Toolkit shall provide the capability to run a single production string in the SCF test environment.

PGSTK-0060 The SCF Environment shall provide the capability for science software developers to generate a production script, capable of linking together multiple PGEs into a single SCF command.

Notes:

A production string may consist of several PGEs run consecutively. The capability to execute such a string from end-to-end will exist at the SCF, through the use of a production script. There will be no capability for executing multiple production strings simultaneously at the SCF.

Requirement:

PGSTK-0080 The SCF Toolkit shall provide the capability to test all I/O transactions among PGEs that originate in the science software, in the same manner as the production environment.

Notes:

As is true of all toolkit functions, input and output function calls in the SCF science software will have the same calling sequences as functions used in the science software at the PGS. The actual physical file locations may be different; for example, instead of a filename in the PGS software that points to a remote location such as another DAAC, the filename will point to a local directory at the SCF. The differences in this case are to be handled through use of Unix environment variables, and so are transparent to the science software. Thus the I/O transaction itself is still being tested as if it is in the production environment.

This means that all SCF I/O interfaces must be identical to the PGS interfaces. The requirement specifically excludes any interaction with the scheduling sub-system, addressed elsewhere in this section. (Section 6.2.1 lists requirements for file I/O tool in the PGS.)

Requirement:

PGSTK-0090 The SCF Toolkit shall contain error and status capabilities identical to the PGS Toolkit and production environment.

Notes:

Another important consideration is to have error and status reporting functions that transform smoothly between the development and production environments. Differences may occur in the destination of error messages, for example; at the PGS these would be sent to the SCF over the net, whereas at the SCF they may go to a file. Again, neither the function calling sequences nor the arguments change; this case may also be handled using Unix environment variables. In cases where it may be necessary to change internal Toolkit code of error and status reporting functions, these changes will be transparent to the user.

It is important that the meaning of error messages be the same in both development and production environment so that causes can be more readily determined.

This requirement implies that the interfaces between these functions and the environment to be identical at both the SCF and the PGS. (Section 6.2.2 lists requirements for error handling in the PGS.)

5.2.3 Code and Hardware

Requirement:

PGSTK-0100 The SCF Toolkit shall contain versions for each of the approved computing platforms.

Notes:

The current list of ECS approved platforms includes: SGI IRIX Indigo R3000, Sun SPARC 10/40, Dec 3000/300, HP 715/50 and IBM RS6000/340 workstations.

Requirement:

PGSTK-0123 SCF Toolkit source code shall be delivered to the SCFs.

Notes:

Access to toolkit source code is essential to science software developers so that they may debug both their own and toolkit software, without having to wait for turnaround at the PGS for toolkit updates. However, developers must be made aware that their code must use the official version of the toolkit software in order to run at the PGS. The DAAC staff will not manage tool software which has been changed at the SCF.

5.2.4 Data Access

Requirement:

PGSTK-0140 The SCF Toolkit shall provide access to Level 0 data provided by science software developers and/or ESDIS.

Notes:

The heart of simulated data in the development environment is the Level 0 data. Such data may be provided by ESDIS, e.g. through the Flight Dynamics Facility, or alternatively by the science software developers themselves.

This requirement addresses only tools that shall access the simulated data; the data itself is assumed to be provided externally.

Science test data formats, which are not addressed by this requirement, are assumed to be identical at both the PGS and the SCF; this to avoid problems at DAAC integration time.

5.2.5 Support and Documentation

Requirement:

PGSTK-0160 The ECS contractor shall provide an Algorithm Support Team at each DAAC, whose function shall be to answer questions about the SCF and PGS Toolkits, PGS design and operations concept, and the science software integration and test process.

Notes:

Ongoing communication between ECS contractor personnel and science software developers is crucial to the success of the SCF development effort, and hence the PGS as a whole; the Algorithm Support Team exists as a bridge between the two groups.

Requirement:

PGSTK-0170 A detailed user's guide for the SCF Toolkit shall be delivered, in both hardcopy and electronic versions, and shall include at least detailed descriptions of the PGS Toolkit; all differences between the SCF and PGS versions, both visible and invisible to Toolkit users, a set of sample production shell scripts; and sample makefiles, including switches for SCF/PGS changes.

5.3 SCF and PGS Environmental Comparison

This section describes the SCF development environment and lists the differences between it and the PGS production environment.

5.3.1 Overview

The details of the PGS environment include how the planner, execution monitor and toolkit functions all interact with each other, as described in Section 3 of this document. In the SCF environment, these components will be emulated to provide the same capabilities as in the PGS environment. However, the implementation of these components may be different in the two environments.

5.3.2 Planner (formerly Scheduler)

PGS—The Planner runs once per day, assembling a list of PGEs to execute based on their priority, along with their file inputs and outputs, and associated resource dependencies; it essentially plans daily activity based on data availability. (See Section 3.3.5 for further details of PGS Planner capabilities.)

SCF—The Planner is not emulated. Since the Planner creates a tentative list of activities that are data driven, the functionality added to the SCF test process would be minimal. Testing of the planning process is best done at integration and test time at the PGS. (Refer to Appendix C for open issues relating to this requirement.)

5.3.3 Execution Monitor

PGS—The Execution Monitor runs in real time; its functions include staging files, monitoring data availability and time constraints, and error handling. It also initiates PGS execution based on data availability, metadata query constraints (such as Q/A flags), computer resources, and priorities.

SCF—The Execution Monitor will not be part of this environment. Instead, its capabilities shall be emulated by the execution of production scripts, which are tailored by the developers for the specific requirements of their science software. (Refer to Appendix C for open issues relating to this requirement.)

5.3.4 Toolkit functions

PGS—These functions, designed for production, are described in Section 7.

SCF—Higher level functions such as math/stat functions and geolocation functions will be identical to the PGS functions. Some lower level functions may have slightly different action, such as error status output, which may be directed to a local file instead of a remote destination, for example. There may also be differences with the PGS version that are transparent to the SCF user. These differences must be kept to an absolute minimum in order to assure smooth integration of science algorithms at the PGS.

In Table 5-1 we show a comparison between the SCF development environment and the PGS production environment.

Table 5-1. A Comparison of SCF and PGS Functionality

Environment Attributes	SCF Development	PGS Production
Resource management	manual	planner
Product generation	production script(s)	execution monitor
Execution criteria	associated table	product orders database/TBD
system resources	identical	
PGE priorities	single PGE	multiple PGEs
metadata constraints	identical	
time thresholds	identical	
data dependencies	identical	
Product strings	single string	multiple strings
Data formats	identical	
File management		
logical locations	mapped by developer	mapped by execution monitor
staging	manual/production-script	execution monitor
Toolkit		
access to level 0 data	simulated (pre-launch) actual (post-launch)	simulated (pre-launch) actual (post-launch)
access to orbit data	simulated (pre-launch) actual (post-launch)	simulated (pre-launch) actual (post-launch)
error/status reporting	local destination	network destination
termination status	handled by production script	handled by execution monitor
I/O interactions	identical	
other tools	identical	
calling sequences	identical	

5.3.5 Other Development Tools

For a discussion of other software development tools such as CASE tools, code checkers, static and dynamic analyzers, and memory monitors, see *Science Users' Guide and Operation Handbook, Vol.4, Algorithm Delivery.*

5.3.6 Science Software Delivery, Integration and Test

Following the development of science software in the SCF environment, the science software should be properly configured for delivery to the DAAC. The delivery procedure should be designed to facilitate the integration and test process which should lead to the eventual transfer of the science software to the operational segment of the DAAC. Each science software package will be subject to three deliveries, plus delta deliveries (within the 6 months preceding launch) to introduce improvements into the operational system. The delivery procedure should require the submission of a packing list, science software design document, all source code, DAAC interface details, science software operations concept document and test specification package. A member of the algorithm support team (AST) will be available, at the DAAC, to assist in the transfer of the science software delivery package to the DAAC. Once the transfer is complete, the algorithm integration and test team (AIT) will subject the science software to the evaluation, integration and acceptance testing phases. Each of these phases will be followed by a review period where the reviewing authority will decide to accept the science software, or return it to the SCF for resolution. Upon successful completion of these reviews, the science software will be transferred to the operational segment of the DAAC. Once here it may undergo additional operational testing on a backup string. Refer to the *Science Users' Guide and Operation Handbook, Vol.4, Algorithm Delivery* for in depth discussion of this topic.

This page intentionally left blank.

6. PGS Toolkit Requirements

In this section, we give a detailed list of requirements for software tools which the EOS science software developers will use to design and test code to be integrated into the PGS environment. These tools will be delivered to the developers so they can emulate the functionality and services of the PGS at their own facilities. The requirements described in this section apply to both the PGS Toolkit as well as the SCF version of the PGS Toolkit. These tools will include both COTS and heritage software as well as custom developed code. They will aid in internal testing of the science algorithms by allowing the use of test data in the same format as that which will be used in the operational environment.

In Appendix A, a matrix of traceability of the requirements in this Section (and Section 5) to Level 3 requirements is given.

At the time of this document, several requirements have been identified which are controversial or are not in the baseline of the present Level 3 Requirements or engineering contract. These are listed in Appendix C.

Appendix D lists potential requirements identified by the science software teams, but not included in Section 6 for various reasons.

A note on the numbering scheme of requirements: Parental (level 3) requirements are identified by the prefix PGS-; Firm PGS Toolkit requirements, i.e. those listed in section 6.2 and 6.3, are identified by the prefix PGSTK-; Potential requirements as given by the science software teams are identified by internal trace number prefix PGS-TR- (in Appendices A, C and D).

6.1 Parental (Level 3) Requirements

We list here those Level 3 Functional Requirements for the ECS which are directly related to the functionality of the PGS Toolkit Requirements Specification proposed in this document. (see Functional and Performance Specification for the Earth Observing System Data and Information System Core System, NAS5-60000, Attachment B, Feb.13,1993, Version).

PGS-0490 The PGS shall have the capability to access and use, for the generation of Standard Products, information such as:

a. Digital terrain map data bases

b. Land/sea databases

c. Climatology databases

d. Digital political map databases

PGS-0510 The PGS shall have the capability to generate metadata (See Appendix C of NAS5-60000, Attachment B) according to the algorithms provided by the scientists and associate this metadata with the Standard Products generated.

PGS-0970 The PGS shall provide file access subroutines that enforce compliance with the adopted standard ECS formats.

PGS-0980 The PGS shall provide job control routines that provide all required task parameters to the Standard Product software.

PGS-0990 The PGS shall provide error logging subroutines for use by Standard Product software in notifying the system operators of conditions requiring their attention.

PGS-1000 The PGS shall provide error logging subroutines for use by Standard Product software in notifying users of conditions requiring their attention.

PGS-1010 The PGS shall provide mass storage allocation subroutines that provide algorithms with a means for dynamic allocation of storage for temporary files.

PGS-1015 The PGS shall provide ancillary data access subroutines that provide Standard Product software access to ephemeris data (i.e., solar, lunar, and satellite ephemeris), Earth rotation data, and time and position measurement data. These subroutines shall perform the following operations:

a. Interpolation

b. Extrapolation

c. Coordinate system conversion.

PGS-1020 The PGS shall provide the following mathematical libraries:

a. Linear algebra and analysis

b. Statistical calculations

PGS-1025 The PGS shall provide a science processing library containing routines such as:

a. Image processing routines

b. Data visualization routines

c. Graphics routines.

PGS-1030 The PGS shall provide a toolkit to the SCF containing versions of the routines specified in requirements PGS-0970 to PGS-1025.

6.2 PGS Toolkit Requirements - System Mandatory

In this section, we list requirements derived from the Level 3 requirements in Section 6.1.

The following three requirements apply to many tools in Section 7.

PGSTK-0180 All PGS Toolkit functions shall return error/status codes that can be detected and reported using error/status reporting tools.

PGSTK-0120 The SCF Toolkit shall be POSIX compliant.

Note: The computing standard to which the PGS Toolkit must adhere is POSIX (Portable Operating System Interface for Computer Environments). Refer to appendix B for a list of PGS approved POSIX calls.

PGSTK-0121 The PGS Toolkit shall provide bindings to ECS approved languages.

Note: The current list of ECS approved languages includes C, Fortran and Ada. Currently, Fortran77 is being supported. However Fortran90 may also be supported when it becomes a POSIX standard on the approved computing platforms.

6.2.1 File I/O Tools

6.2.1.1 Level 0 Science Data Access

PGSTK-0190 The PGS Toolkit shall contain tools to open and close Level 0 data sets.

PGSTK-0200 The PGS Toolkit shall contain tools to read Level 0 data for a specified time or orbit location.

PGSTK-0210 The PGS Toolkit shall contain tools to access assembled Level 0 granules and validation flags.

PGSTK-0220 The PGS Toolkit shall provide access to the end of the prior Level 0 granule and the beginning of the next Level 0 granule, concurrently with the present Level 0 granule. The precise amount of end-data is TBD.

PGSTK-0230 The PGS Toolkit shall contain tools to access the Level 0 metadata, i.e. EDOS accounting.

PGSTK-0240 The PGS Toolkit shall provide access to CERES and LIS Level 0 data from TRMM.

6.2.1.2 HDF File Access

PGSTK-0271 The PGS Toolkit shall contain tools that check the contents of HDF files, i.e., what HDF data type and number of items it contains.

PGSTK-0270 The PGS Toolkit shall contain tools that read and write HDF files.

PGSTK-0250 The PGS Toolkit shall provide HDF file access tools which include as inputs at least run ID and standard product ID codes.

PGSTK-0290 The PGS Toolkit shall contain tools to read from and write to an HDF file, the fundamental quantities associated with a data set.

Note:

Refer to section 7.2.1.2 for examples of fundamental quantities.

PGSTK-0300 The PGS Toolkit shall contain tools to read from and write to an HDF file (1) an entire data set at once, (2) a fraction of a data set and (3) an N-dimensional data array. The HDF file may contain multiple data sets.

PGSTK-0320 The PGS Toolkit shall contain tools to read from and write to an HDF file, variable length sub-records.

PGSTK-0310 The PGS Toolkit shall have the capability to notify SCFs of the creation of HDF files.

6.2.1.3 Non-HDF File Access

PGSTK-0360 The PGS Toolkit shall contain tools to open, read, write, and close non-HDF files, including quicklook, text and binary files.

6.2.1.4 Metadata

PGSTK-0370 The PGS Toolkit shall support opening a metadata file.

PGSTK-0380 The PGS Toolkit shall be able to write a header to a metadata file containing standard and science-software-specific information.

PGSTK-0390 The PGS Toolkit shall be able to position the insertion point in the metadata file.

PGSTK-0400 The PGS Toolkit shall be able to write a record of metadata in the metadata file using ECS standard structuring, and contain ECS standard, instrument specific and product specific attributes. The record will contain program variables and constants as well as values generated automatically (e.g. configuration information)

PGSTK-0410 The PGS Toolkit shall be able to overwrite a record in a metadata file with a new record.

PGSTK-0440 The PGS Toolkit shall be able to write a terminator to a metadata file.

PGSTK-0430 The PGS Toolkit shall support closing a metadata file.

PGSTK-0450 The PGS Toolkit shall support writing the ECS standard, instrument specific and product specific attributes into the ECS standard product file.

6.2.1.5 Data Quality Assurance

PGSTK-0490 The PGS Toolkit shall contain tools to append data quality assurance information to an HDF file.

PGSTK-0500 The PGS Toolkit shall contain tools to pass flags which can be used to notify the SCF that Q/A data at the PGS is complete, and notify the DAAC service (TBD) to automatically ship the file to the SCF if requested.

PGSTK-0510 The PGS Toolkit shall contain tools that support three types of Q/A data: (1) flags; (2) graphics files, which are output directly from science processes; and (3) data which is written in the same manner as a standard product file.

6.2.1.6 Temporary Files

PGSTK-0520 The PGS Toolkit shall contain tools that delete PGE specific temporary files when the PGE terminates.

PGSTK-0540 The PGS Toolkit shall contain tools that reverse the deletion process for temporary files.

PGSTK-0530 The PGS Toolkit shall contain tools that ensure that temporary file names do not conflict with any other file names.

PGSTK-0531 The PGS Toolkit shall contain tools that access files which are not standard products, but are required to be available for up to 1 months time.

PGSTK-0532 Provide a tool to perform deletion of a temporary file that is shared among PGEs.

PGSTK-0533 Provide a means to save temporary files over successive executions of a program.

PGSTK-0534 Provide access to temporary calibration files from the last orbit.

6.2.1.7 Miscellaneous I/O

PGSTK-0550 The PGS Toolkit shall contain tools to generate ECS standard file names dynamically, which include science software program identification ID, file version ID, and are independent of computer platform. These will be used as logical file names within a science process.

PGSTK-0570 The PGS Toolkit shall contain tools to monitor for the presence of the quicklook flag, which indicates that transmission to the SCF of part of the data from the spacecraft or from Level 1, is required.

6.2.2 Error/Status Reporting

PGSTK-0580 The PGS Toolkit shall contain tools to interpret and report multi-level error/status conditions to the PGS and the SCF.

PGSTK-0590 The PGS Toolkit shall contain tools to report fatal errors, warning errors, informational messages, notices and user-defined status events.

PGSTK-0600 The PGS Toolkit shall contain tools to report error and status conditions to the PGS scheduler, PGS monitoring and accounting, PGS processing logs, PGS operators, the PGS Q/A analysts, and the SCFs.

PGSTK-0610 The PGS Toolkit shall contain tools to uniquely identify the software unit, science software program, product and production run in error and status messages.

PGSTK-0620 The PGS Toolkit shall contain tools to identify the associated instrument within the error message codes.

PGSTK-0630 The PGS Toolkit shall contain tools to send all temporary files, processing logs and error logs to the SCF, when processing errors occur during execution of the current PGE that cannot be resolved by PGS personnel.

PGSTK-0650 The PGS Toolkit shall contain tools to associate with error messages at least the following: what routine noted the error, error-type, pertinent variable data, and action taken.

6.2.3 Process Control

PGSTK-1280 The PGS Toolkit shall provide tools to generate product identifiers which can be used by the script/PGE to label metadata with environment and PGE information in order to facilitate production tracking.

PGSTK-1290 The PGS Toolkit shall provide tools to deliver runtime configuration parameters to the PGE. Examples of these parameters include: executable mode values, availability of alternate data files and system defined directory paths for temporary files.

PGSTK-1310 The PGS Toolkit shall provide the capabilities to query for availability and validity flags.

6.2.4 Memory Management

PGSTK-1240 The PGS Toolkit shall provide tools which (1) dynamically allocate process-private memory (perhaps with limits) and (2) explicitly free dynamic memory within a program when it is no longer needed.

6.2.5 Ancillary Data Access and Manipulation

PGSTK-1360 The PGS Toolkit shall provide access interfaces to all ancillary data sets supported by the ESDIS Project. The exception to this are the static files from internal sources which will be supported through the standard I/O functions within the toolkit and will not, therefore, need formal requirements.

PGSTK-1361 Configuration information about data generation shall be supplied to the calling routines.

6.2.5.1 Dynamic ancillary data from internal sources

Note: Dynamic data sets are those containing parameters whose values change routinely

and predictably; i.e. at set intervals in time.

PGSTK-0730 The PGS Toolkit shall contain tools which return times of significant spacecraft events which may disturb spacecraft pointing within a given time range.

PGSTK-1251 The PGS Toolkit shall provide an interface to products from other EOS instruments in ECS standard formats.

PGSTK-1252 The PGS Toolkit shall permit AIRS level 1b products to be used as ancillary inputs for the MOPITT level 2 processing.

PGSTK-1253 The PGS Toolkit shall permit AIRS level 2 products to be used as ancillary inputs for the ASTER, CERES, MISR, MODIS and MOPITT level 2 and LIS level 1b processing.

PGSTK-1254 The PGS Toolkit shall permit AMSU level 1b products to be used as ancillary inputs for the AIRS and MOPITT level 2 processing.

PGSTK-1255 The PGS Toolkit shall permit ASTER level 1b products to be used as ancillary inputs for the MOPITT level 2 processing.

PGSTK-1256 The PGS Toolkit shall permit ASTER level 2 products to be used as ancillary inputs for the MISR level 2 processing.

PGSTK-1257 The PGS Toolkit shall permit CERES level 2 products to be used as ancillary inputs for the MISR and MOPITT level 2 processing.

PGSTK-1258 The PGS Toolkit shall permit EOSP level 2 products to be used as ancillary inputs for the MISR level 2 processing.

PGSTK-1259 The PGS Toolkit shall permit MIMR level 1b products to be used as ancillary inputs for the CERES level 2 processing.

PGSTK-1260 The PGS Toolkit shall permit MIMR level 2 products to be used as ancillary inputs for the LIS level 1b and MODIS level 2 processing.

PGSTK-1261 The PGS Toolkit shall permit MISR Level 2 products to be used as ancillary inputs for the ASTER and MODIS level 2 processing.

PGSTK-1262 The PGS Toolkit shall permit MODIS level 1b products to be used as ancillary inputs for the CERES and MOPITT level 2 processing.

PGSTK-1263 The PGS Toolkit shall permit MODIS level 2 products to be used as ancillary inputs for the LIS level 1b and MISR level 2 processing.

PGSTK-0925 The PGS Toolkit shall provide an interface to latent heating profile data values from a data set whose origin is TRMM/NSSDC .

6.2.5.2 Static ancillary data from internal sources

Note: Static data sets are those containing parameters whose values may change, but not routinely at set intervals in time.

PGSTK-1265 The PGS Toolkit shall provide a means to access static ancillary data in ECS standard format data files associated with particular algorithms.

6.2.5.3 Static auxiliary data from external sources

PGSTK-0840 The PGS Toolkit shall provide a means to retrieve the required geophysical parameter from the data set at a specified location.

PGSTK-0850 The PGS Toolkit shall provide a means to retrieve regular grids or volumes of the required geophysical information defined by the upper left and bottom right vertices (x,y,z at each vertex in a supported geographic coordinate system).

PGSTK-0870 The PGS Toolkit shall contain tools to access a land/sea classification database including coastal outlines.

PGSTK-0980 The PGS Toolkit shall provide a means to retrieve elevation and terrain information from the DTM (slope and aspect) at a specified latitude and longitude coordinate.

PGSTK-1000 The PGS Toolkit shall provide a means to receive from the DTM a regular grid of elevation, slope and aspect information for a rectangular area defined by the maximum extent of the rectangle.

PGSTK-1030 The PGS Toolkit shall contain tools to provide one-sigma uncertainty on output parameters (elevation, slope, aspect values) for output of digital elevation and terrain access tools.

PGSTK-1040 The PGS Toolkit shall contain a tool that provides digital elevation model data in Space Oblique Mercator coordinates, and includes the minimal elevation within a 10 km radius.

PGSTK-1070 The PGS Toolkit shall provide the geoid to ellipsoid difference at a specific latitude/longitude coordinate.

6.2.5.4 Dynamic auxiliary (external) data

PGSTK-0921 The PGS Toolkit shall provide an interface to retrieve cloud values from a data set whose origin is NOAA/NESDIS.

PGSTK-0922 The PGS Toolkit shall provide an interface to retrieve GMS radiance values from a data set whose origin is GMS/NASDA.

PGSTK-0923 The PGS Toolkit shall provide an interface to retrieve GOES radiance values from a data set whose origin is GOES/NESDIS.

PGSTK-0924 The PGS Toolkit shall provide an interface to retrieve GOES VAS data values from a data set whose origin is GOES/NESDIS.

PGSTK-0926 The PGS Toolkit shall provide an interface to retrieve sea level pressure analysis data values from a data set whose origin is in situ measurements made by NMC.

PGSTK-0927 The PGS Toolkit shall provide an interface to retrieve sea surface temperature values from a data set whose origin is NOAA/NESDIS.

PGSTK-0928 The PGS Toolkit shall provide an interface to retrieve surface wind values from a data set whose origin is DMSP/NMC.

PGSTK-0929 The PGS Toolkit shall provide an interface to retrieve ozone values from a data set whose origin is GOMR .

PGSTK-0931 The PGS Toolkit shall provide an interface to retrieve values from a data set whose origin is ECMWF .

PGSTK-1370 The PGS Toolkit shall provide a tool to interpolate auxiliary data such as NMC data in time and space.

6.2.6 Time and Date Conversion Tools

PGSTK-1170 The PGS Toolkit shall provide tools to transform time among the four following systems:

a. spacecraft clock

b. UTC (binary and ASCII formats)

c. ephemeris time (binary and ASCII formats)

d. Julian date (binary and ASCII formats).

PGSTK-1215 The PGS Toolkit shall provide a tool to convert ephemeris times from one ephemeris reference epoch to another.

PGSTK-1220 The PGS Toolkit shall provide a tool to convert Julian dates from one Julian reference epoch to another, with a precision of 9 significant digits.

PGSTK-1160 The PGS Toolkit shall contain time system transformation tools that return UTC and ephemeris times and Julian dates that are of the same precision as the spacecraft clock.

PGSTK-1180 The PGS Toolkit time system transformation tools shall return times that are in Consultative Committee for Space Data Systems (CCSDS) standard time code formats.

PGSTK-1190 The PGS Toolkit time system transformation tools shall have the capability of returning ephemeris time in seconds from the start of a specified epoch.

PGSTK-1195 The PGS Toolkit time system transformation tools shall have the capability of returning Julian dates in seconds from the start of a specified epoch.

PGSTK-1200 The PGS Toolkit time system transformation tools shall, at a minimum, return times of one millisecond resolution.

PGSTK-1210 The PGS Toolkit shall assure that leap seconds are accounted for in all time and date conversion tools.

PGSTK-1211 The PGS Toolkit shall provide a tool to access the spacecraft time difference file for foreign platforms.

6.3 PGS Toolkit Requirements - Science

6.3.1 Heavenly Body and Spacecraft Position Access Tools - Mandatory

PGSTK-0680 Input to all relevant PGS Toolkit planetary body and spacecraft position access functions shall include spacecraft identification.

PGSTK-0750 Geographic information access tools in the PGS Toolkit that use latitude as a parameter shall also use co-latitude as a parameter.

PGSTK-0930 Geographic information access tools in the PGS Toolkit shall be capable of handling the north and south pole singularities, e.g. by increasing resolution by a TBD prescription.

Note: The requirements above are general ones which apply to many of the tools in Section 7.3.1 .

6.3.1.1 Spacecraft Ephemeris & Attitude Data Access Tools

PGSTK-0710 The PGS Toolkit shall use a single standard time interval (which is TBD) in all ephemeris calculations.

PGSTK-0720 The PGS Toolkit shall provide tools to return spacecraft position, velocity and attitude for any given time or for a range of times, including provision for interpolation between state vectors.

PGSTK-0740 The PGS Toolkit shall contain tools which return a flag indicating whether reported attitude and position is reliable within a pre-existing (user supplied) specification.

6.3.1.2 Heavenly Body Position Access Tools

PGSTK-0760 The PGS Toolkit shall contain tools that return local solar time, and solar right ascension and declination for a given time and position on the Earth's surface.

PGSTK-0860 The PGS Toolkit shall contain a tool to determine if a given point on the earth's surface is in day or in night.

PGSTK-0770 The PGS Toolkit shall contain tools that return Greenwich Hour Angle for a given time and position on the Earth's surface.

PGSTK-0900 The PGS Toolkit shall contain tools to determine the inertial location of the Greenwich meridian.

PGSTK-0790 The PGS Toolkit shall contain tools that provide access to planetary ephemerides and star locations.

PGSTK-0780 The PGS Toolkit shall contain tools that return a flag for a disturbance event caused by a celestial body.

PGSTK-0820 The PGS Toolkit shall contain a tool that returns positions of stars of luminosity greater than a magnitude which is TBD.

PGSTK-0800 The PGS Toolkit shall contain a tool that returns the Earth-Centered Inertial (ECI) vector from the Earth to the Sun, Moon, and planets at a given time.

6.3.1.3 Coordinate System Conversion

PGSTK-1050 The PGS Toolkit shall provide the following lower level coordinate system transformations:

a. spacecraft reference to orbital reference

b. orbital reference to Earth reference ellipsoid

c. Earth reference ellipsoid to spacecraft reference

d. Earth reference ellipsoid to Space Oblique Mercator (SOM)

e. Earth-Centered Inertial (ECI) to Earth-Centered Rotating (ECR)

f. ECR to geodetic

g. ECI to spacecraft reference

h. each coordinate transformation will be bi-directional

PGSTK-1080 The PGS Toolkit shall provide the intersection of the instrument look vector in the spacecraft reference frame with the earth reference ellipsoid at an arbitrary time.

PGSTK-1060 The PGS Toolkit shall provide the sub-satellite point and ground track velocity vector at any arbitrary time.

6.3.1.4 Geo-Coordinate Transformation

PGSTK-1500 The PGS toolkit shall support the bi-directional transformation between coordinates in the Cartesian ellipsoid reference frame and the Space Oblique Mercator, Universal Transverse Mercator, Polar Stereographic, and the Goodes Interrupted Homolosine Projections

PGSTK-1501 The PGS Toolkit shall provide a tool to transform coordinates from Universal Transverse Mercator to Space Oblique Mercator.

PGSTK-1502 The geo-coordinate transformation tools shall support the transformation of multiple coordinate vectors in a single call.

6.3.2 Math & Modeling Support Tools - Optional

PGSTK-1245 The PGS Toolkit shall provide tools to accomplish various mathematical and statistical tasks including, but not limited to:

a. Solution of linear algebraic equations, matrix manipulation, matrix inversion and Eigenvalue decomposition.

b. Interpolation and extrapolation

c. Integration and evaluation of functions

d. Root finding

e. Determination of min/max of functions

f. Statistical description of data

g. Fast Fourier Transforms and polynomial fits.

6.3.3 Constants and Unit Conversions - Optional

6.3.3.1 Math, Physical and Instrument Constants

PSGTK-1520 The PGS Toolkit shall provide a means of accessing commonly used mathematical and physical constants.

PGSTK-1521 The PGS Toolkit shall provide a means of accessing constant values related to an instrument.

PGSTK-1522 The values of the parameters shall be capable of adjustment without recompilation of a PGE.

6.3.3.2 Unit Conversions

PGSTK-1530 The PGS Toolkit shall provide a means to perform unit conversions and parameter translations.

PGSTK-1531 The unit conversion tools shall transform multiple values in a single call.

6.3.4 Graphics Support Tools - Optional

PGSTK-1410 The PGS Toolkit shall provide tools for producing graphics output from production software.

PGSTK-1415 The PGS Toolkit shall provide tools for image processing; for map projections; correlations and registration; filters; contrast enhancement. The PGS Toolkit shall provide tools to compute statistics (mean, variance, etc.) across pixels in a contiguous or non-contiguous area of interest of an image.

7. PGS Toolkit Specification

7.1 Introduction

In this section, we give a preliminary list of PGS Toolkit software tools which are allocated to the requirements in Section 6 of this document. The following fields are provided: a name, a description of each tool, a list of input and output files and parameters, a cross reference to the target Toolkit requirement(s) and a synopsis field (which in the present preliminary version is left blank). This Section will grow into a design document which will be contained in the initial Toolkit delivery. The design document will contain calling sequences and interface description of all tools. See Section 8 for a proposed delivery schedule of Toolkit routines.

In this Section, it is assumed that ECS science software requests for system services, for system and resource accesses, file I/O requests, error message transaction, metadata formatting, and geographic information data base requests must be made through the Toolkit. Furthermore, this usage will be tested at integration time at the DAACs. A listing of prohibited and allowed POSIX calls is provided in Appendix B. Other services such as scientific and math library calls, will be provided by the Toolkit. However, in these latter cases, usage will not be enforced, but encouraged.

In this section, we provide a list of tools allocated to the requirements in Section 6.

7.2 PGS Toolkit Tools - System Mandatory

7.2.1 File I/O Tools

This section describes the set of tools used to perform file I/O, including L0, HDF, and non-HDF (e.g. binary files) format data.

Before getting into specific tool descriptions, we discuss the method by which the science software will obtain file name identifiers from the PGS system.

The basic file I/O concept as provided by the Toolkit is based on programmer supplied logical names. These logical names are bound to physical files at runtime through the PGS/SCF environments. The exact method of binding depends on the file type and the mode of operation (production vs. development/test).

For example, in production mode, product, ancillary, and auxiliary files are staged by the schedule execution subsystem (SES). The SES uses parameters provided by algorithm developers (through the algorithm integration facility) to obtain the necessary data from the IMS/DADS and bind it to the supplied logical name. The science algorithm will need to be constructed to use this logical name in file open calls, and the PGS operational environment will provide the runtime binding to support file access. Because the SES uses data availability in determining algorithm "readiness", dataset synchronization is handled external to PGE execution, and file open calls should proceed unblocked.

In development and integration mode, this logical to physical binding is provided via a Unix-like "environment", which provides name-value pairs for resolving the logical-to-physical mapping.

Temporary files (ones that don't require product generation and archiving) also use a logical naming scheme. In this situation, the first open call will create a physical file bound to the supplied logical name. Future references to that logical name will provide access to the newly created file.

7.2.1.1 Level 0 Science Data Access Tools

This section describes tools which access raw science data contained in the Level 0 data stream supplied by EDOS. In the description below it is assumed that a granule is a minimal block of Level 0 data which can initiate a Level 1 science production process. Spacecraft ancillary engineering data access tools are described in Section 7.2.5.

Tool Name:	**Open L0 File**
Description:	This tool opens Level 0 data sets.
Requirements:	PGSTK-0190
Inputs:	Filename
Outputs:	File handle
	Status return value
Synopsis:	

Tool Name:	**Close L0 File**
Description:	This tool closes Level 0 data sets.
Requirements:	PGSTK-0190
Inputs:	File handle
Outputs:	Status return value
Synopsis:	

Tool Name:	**Read L0 Granule**
Description:	This tool is used to read Level 0 granules. It will include data from the end of the previous granule and the beginning of the next granule, the amount from the ends determined by a delta time.
Requirements:	PGSTK-0200, 0210, 0220
Inputs:	File handle
	Time
	Delta time: start/stop time
	Ephemeris data
Outputs:	Level 0 data
	Validation data
	Status return value
Synopsis:	

Tool Name:	**Read L0 EDOS Accounting**
Description:	This tool handles the reading of Level 0 EDOS accounting information. This includes such fields as file size, file type (Level 0 data) and the number of packets in the file. The header will vary depending on whether the data is regular, Level 0 quicklook, or Level 0 retrieved from the EDOS archive.
Requirements:	PGSTK-0230
Inputs:	File handle from open call
	Information type (code)
Outputs:	Requested information
	Status return value
Synopsis:	

Tool Name:	**Read L0 TRMM File**
Description:	This tool reads Level 0 data from TRMM format data files.
Requirements:	PGSTK-0240
Inputs:	File handle
	Start/stop time
Outputs:	L0 data
	Validation data
	Status return value
Synopsis:	

7.2.1.2 HDF File Access Tools

This section lists tools which meet the requirements to read and write HDF (Hierarchical Data Format) files, which is the ECS baseline standard file data format. An introductory description of HDF is given in *Getting Started with HDF,* U. of Illinois, 1993.

It is assumed that HDF will include read and write access to netCDF, that netCDF has been incorporated into HDF.

Tool Name:	**Inspect HDF File**
Description:	This tool checks the contents of an HDF file.
Requirements:	PGSTK-0271
Inputs:	File handle

Outputs: Data type(s) in the HDF file (RIS8, RIS24, SDS, Vset)

Number of data items for each data type

Status return value

Synopsis:

Tool Name: **Get HDF Data Structure Info**

Description: This tool provide detailed information about the structure of data items inside an HDF file.

Requirements: PGSTK-0271

Inputs: File handle

Data type (RIS8, RIS24, SDS, Vset)

Item selection (first, current, specific number)

Outputs: Size and structure of data item (image size and type, presence and type of image palette, type and size of SDS, size of Vset - indication if Vset contain Vset data type)

Status return value

Synopsis:

The following group of tools address the general read and write requirements for HDF file access routines. These routines will access HDF files of Level 1-4 and browse data, and support the writing of quicklook data to HDF files. Detail is provided about the types and sizes of data items to be transferred during read and write operations.

Tool Name: **Read HDF File**

Description: This tool will read HDF files to allow access to Level 1-4 and browse products.

Requirements: PGSTK-0250, 270, 290, 300, 320

Inputs: File handle

Type of item (image, data set, partial data set, annotation)

Destination of read operation, conversion type (destination format)

Number of items

Outputs: Requested data (data sets, partial data sets, run ID, standard product ID)

Status return value

Synopsis:

Tool Name: **Write HDF File**

Description: This tool will write to HDF files, to allow storing of Level 1-4, browse, and quicklook products.

Requirements: PGSTK-0250, 0270, 0290, 0300, 0320

Inputs: File handle, access mode (write, append, update), run ID, standard product ID

Type of item (data set, partial data set, array, Vset, storage format)

Items properties (size, rank, format)

Data items

Outputs: Status return value

Synopsis:

Tool Name: **Select HDF Data Item**

Description: This tool selects a data item in an HDF file which contains multiple data items of the same type.

Requirements: PGSTK-0250, 0270, 0290, 0300, 0320

Inputs: File handle

Type of item (data set, partial data set, array, Vset)

Data item selector (first, next, item number, search for ID)

Outputs: Item number, (sets HDF toolkit item indicator to selected item)

Status return value

Synopsis:

Tool Name: **Send Product Availability Notice**

Description: This tool provides a mechanism to notify personnel at the SCF that a data product is ready.

Requirements: PGSTK-0310

Inputs: Identification of new HDF file (filename, annotation, contents list, date and time)

Outputs: Status return variable

Synopsis:

7.2.1.3 Non-HDF File Access Tools

These tools shall be wrappers around standard language-specific file I/O routines, including as additional functionality such items as directory and file name concatenation, for example.

Tool Name: **Open Non-HDF File**

Description: This tool opens a non-HDF file.

Requirements: PGSTK-0360

Inputs: File name (assumed to be a logical name)

Access mode (read only, write only, read/write, append)

Outputs: File handle

Status return variable

Synopsis:

Tool Name: **Close Non-HDF File**

Description: This tool closes a non-HDF file.

Requirements: PGSTK-0360

Inputs: File handle

Outputs: Status return variable

Synopsis:

Tool Name: **Get Formatted Data from Non-HDF File**

Description: This tool reads from a non-HDF file and converts output to a specified format.

Requirements: PGSTK-0360

Inputs: File handle

Data input (quicklook file, text file, binary file, image)

Conversion type (input format, output format)

Number of items

Location in program memory where data items shall be stored

Outputs: Data output (stored in program memory)

Status return variable

Synopsis:

Tool Name: **Write Non-HDF File**

Description: This tool writes to a non-HDF file.

Requirements: PGSTK-0360

Inputs: File handle

Data input (arrays, variables from program memory)

Type (character, string, record, binary), size of data item

Number of items

Outputs: Data output (quicklook file, text file, binary file)

Status return variable

Synopsis:

7.2.1.4 Metadata

Tool Name: **Access Metadata File**

Description: The association of metadata with scientific data sets is essential to promote the long-term usage of the data set in order to facilitate parameter-based data searches which identify those parts of a data set that have characteristics relevant to a user query. The scope and long-term objectives of the EOS mission necessitate full documentation for every ECS-standard product.

Several forms of metadata will exist in the ECS. Metadata will be included in the product itself as a permanent descriptor or the product. A subset of the metadata in the product will also be loaded in the IMS. This will be achieved through the generation of a metadata file during science processing which can be automatically loaded into the IMS

The metadata attributes will be partially defined by IMS requirements and partly by the science requirements. It is assumed that there will be a standard set of ECS metadata attributes, some instrument specific parameters and some product specific attributes. The current baseline attributes are listed in Tables C-10 and C-11 of the Functional and Performance Requirements Specification.

A science algorithm will use the PGS toolkit to place the required attributes into a standard record structure for output to a metadata file or for input into the ECS standard format product itself. The required content will be the standard ECS metadata attributes and a significant amount of configuration information relating to the ancillary data files used in the processing to provide an audit trail. This configuration information will be generated automatically and will not require parameters to be generated from within the algorithm code.

The metadata file will have a defined format and the toolkit will support the initialization, creation and termination of this type of file. The output metadata files are intended for ingestion into the IMS, but will not be passed to the IMS by the PGS toolkit; this process will occur at the end of processing. The PGS toolkit will also support updating of records written to the metadata file.

Requirements: PGSTK-0370,0380,0390,0400,0410,0430,0440,0450

Inputs: type of action (open, write, update, position, terminate, close); destination (metadata file, ECS standard product file); program variables/constants

Outputs: complete record for writing to metadata file or ECS standard product file; metadata file; error/status

Synopsis:

Notes: The design for this set of tools can only be established once the standards for metadata attributes, and IMS update procedures have been defined. Since this definition is likely to evolve, early releases of the toolkit may not produce the metadata files in the final format to be used for ECS. Output of metadata to an ECS standard product file will use the PGS Toolkit functions for writing to such a file.

7.2.1.5 Data Quality Assurance

The nature of the data quality assurance (Q/A) information, and the means of its generation, is expected to vary significantly among products. The tools proposed here will not themselves evaluate Q/A information, but will supply means for writing that information.

There is no requirement for tools to read Q/A data files, as that is the province of the science software. Q/A flags and graphics files will be supported by read tools; however, see Section 7.3.4.

Tool Name: **Send Q/A Flags**

Description: This tool will output Q/A flags to either a file (append), the SCF or the DAAC.

Requirements: PGSTK-0490, 0500, 0510

Inputs: Destination type (file append | SCF notify | DAAC notify)

Destination specification (file handle| SCF address | DAAC address)

Q/A flags

Outputs: Status return value

Synopsis:

Tool Name: **Write Q/A to Graphics File**

Description: This tool will write Q/A information to a graphics file.

Requirements: PGSTK-0490, 0510

Inputs: File handle

Q/A data

Outputs: Status return value

Synopsis:

Tool Name: **Write Q/A to HDF File**

Description: This tool will write Q/A information to a standard products file.

Requirements: PGSTK-0490, 0510

Inputs: File identification

	Q/A flags
Outputs:	Status return value
Synopsis:	

7.2.1.6 Temporary Files

A temporary file is a file which may exists for the duration of a single PGE, or may exist for some indeterminate time beyond the termination of the PGE which created it. Normally this file will be deleted following successful termination of a PGE. However, in some cases it may be advantageous for future processing to utilize this temporary data.

All tools referenced above in the section entitled "Non-HDF Files" will be available for the user to access a temporary file. Additionally, the following tools provide for more esoteric file operation requirements.

Tool Name:	**Delete Temporary File**
Description:	This tool will allow the deletion of a temporary file.
Requirements:	PGSTK-0520
Inputs:	File identification
Outputs:	Status return value
Synopsis:	

Tool Name:	**Create Temporary File Name**
Description:	This tool will generate a unique temporary file name.
Requirements:	PGSTK-0530, 0540
Inputs:	Duration
Outputs:	File name (assumed logical)
	Status return value
Synopsis:	
Notes:	The duration of the temporary file will default to the PGE life span, unless other directives are given. These directives may be in the form of a number of days, or a duration model (i.e. SHORT, MEDIUM, LONG).

The PGS system may impose restrictions on the number and/or size of temporary files that a PGE can generate, and on the duration of their existence at the DAAC.

7.2.1.7 Miscellaneous Tools

The tools described in this section include file-naming tools and quicklook file send tools.

Tool Name:	**Create Standard File Name**
Description:	This tool will generate an ECS standard file name.
Requirements:	PGSTK-0550
Inputs:	Science software program identification ID File version ID Date Time
Outputs:	ECS standard file name Status return value
Synopsis:	

Tool Name:	**Repackage Quicklook Data**
Description:	This tool will repackage an indicated part of the L0 or L1 data stream for transmission to the SCF as quicklook data.
Requirements:	PGSTK-0570
Inputs:	File handle for L0 or L1 data
Outputs:	Status return value
Synopsis:	

7.2.2 Error/Status Reporting Tool Requirements

The purpose of these tools is to propagate status information within a PGE executable to facilitate the error handling process and the communication of status information to various monitoring authorities and event logs.

In order to detect and report on error and status conditions in a consistent manner, standardized status messages and status codes must first be established. These status codes will be unique across the entire PGS system to ensure that there will be no ambiguous status conditions. In fact, the PGS Toolkit routines will contain a collection of status codes and associated status messages for describing the state of each Toolkit routine. In order to inform a calling unit about its exit state, each Toolkit routine will set a status message, and assign a status code to the return value. Based on its interpretation of this return value, the calling unit may elect to perform some error handling. As part of this procedure, the user should be able to either propagate the existing status code up the calling hierarchy, or set a more informative status code and message to represent the outcome of any error handling attempt.

Upon detection of an error state, it will be advisable to report on the existing error prior to performing an error handling procedure. The content of these reports may include a user-defined message string, additional information which uniquely identifies the environment in which the error occurred, as well as information relating to an action about to be taken in response to the error condition. The method for reporting this information will involve the retrieval of a pre-set status message and subsequent transmission of that message to the appropriate destinations.

Tool Name:	**Set Static Status Message**

Description: This tool will provide the means to set a user-defined error/status code and message in response to the outcome of some segment of processing.

Inputs: Mnemonic error/status code

Outputs: none

Requirements: PGSTK-0580, 0590

Synopsis:

Notes: This tool should be used within the error handling section of a program module. The return value from the module should be the same as the last status code that was set in that module. This will enable calling modules to retrieve the status message set in the previously called module.

Tool Name: **Set Dynamic Status Message**

Description: This tool will provide the means to set user-defined error/status code and message in response to the outcome of some segment of processing. Additionally, this tool will accept variable runtime information, to be embedded within the associated message string. The number of variables is only limited by the number indicated by the message template. The message template may contain variable place holders which get substituted at runtime through the use of this tool.

Inputs: Mnemonic error/status code

Variable number of runtime values

Outputs: none

Requirements: PGSTK-0580, 0590, 0650

Synopsis:

Notes: This tool should be used when setting an error status in order to incorporate runtime variables that are currently in scope (i.e. local variables in the called module).

Tool Name: **Set Error Action**

Description: This tool will provide the means for the user to attach a string to an error message to indicate an action that will be taken in response to an existing error condition. This tool should be used prior to the error reporting and exception handling phases normally associated with an error condition.

Inputs: Mnemonic error/status code

String specifying action taken.

Outputs: none

Requirements: PGSTK-0580, 0590, 0650

Synopsis:

Notes: The action string represents static information to be passed along during status reporting.

Tool Name: **Get Static Status Message**

Description: This tool will provide the means to retrieve a pre-defined message string that is associated with the error/status code.

Inputs: Mnemonic error/status code

Outputs: Associated message string

Requirements: PGSTK-0580, 0590, 0600

Synopsis:

Tool Name: **Get Dynamic Status Message**

Description: This tool may be used to generate a runtime specific message from a pre-defined message template. The number of variables is only limited by the number indicated by the message template. The message template may contain variable place holders which get substituted at runtime through the use of this tool.

Inputs: Mnemonic error/status code

Variable number of runtime values

Outputs: Associated message string containing runtime values

Requirements: PGSTK-0580, 0590, 0600

Synopsis:

Notes: This tool should be used when retrieving an error message, in order to incorporate runtime variables that may not have been in scope at the time the error status was set (i.e. local variables in the calling module).

Tool Name: **Send Status Report**

Description:

This tool may be used to effect the transfer of error/status messages to key monitoring authorities during a production run.

Inputs: Complete error/status message string

List of network aliases

Outputs: Status return value

Requirements: PGSTK-0580,0590,0600

Synopsis:

Notes: It is assumed that the Toolkit will have access to network destination addresses mapped to alias names (for example, by a CSMS directory service). The Toolkit can access this file prior to transmission of a status report.

Tool Name: **Create Message Tag**

Description: This tool may be used to generate a unique message identifier. This identifier may be attached to error/status message strings to facilitate tracking of error/status messages to a module of code within a product executable, for a unique production run.

Inputs: Production run id

Product id

Science software program id

Software Unit id

Outputs: Unique message identifier label

Requirements: PGSTK-0610

Synopsis:

Notes: If more than one product is being generated from the same PGE, then the appropriate Product Id must be used as input to this routine when called from within the science software.

Tool Name: **Get Instrument Name**

Description: This tool may be used to recover the instrument name from an error/status code.

Inputs: String representation of mnemonic status code

Outputs: Instrument name

Requirements: PGSTK-0620

Synopsis:

Notes: The user will construct the error/status codes in a manner which incorporates, as a prefix, a pre-defined instrument code.

Tool Name: **Send Runtime Data**

Description: This tool will provide a means for the user to transmit a package of runtime data to the SCF in the event of an unhandled system exception. This package may contain status/error logs and volatile temporary files that may be useful for discerning the problem which may have initiated the exception. This tool may only be invoked once for a given process.

Inputs: List of temporary files & error logs

Network alias of SCF

Outputs: Status return value

Requirements: PGSTK-0630

Synopsis:

Notes: It is assumed that the Toolkit will have access to network destination addresses mapped to alias names (for example, by a CSMS directory service).

Tool Name: **Generate Status Report**

Description: As part of the error/status log file update, this tool will invoke a routine to generate a message identifier which will be incorporated into the message string being written to the file. Once the complete message string is assembled, it will be written to the specified error/status log file.

Inputs: Name of error/status log file

Error/status message string

Production run id

Product id

Science software program id

Software Unit id

Outputs: Status return value

Requirements: PGSTK-0650

Synopsis:

Notes: Information regarding error type, variable data, and action taken should already be part of the input message string.

7.2.3 Process Control Tools

The tools described in this section will be used primarily in the early stages of a PGE to refine the characteristics if a particular production run.

Examples are tools which: allow runtime information (e.g. dimensions of input data sets) to be passed to the PGE from the scheduling system, generate identifier labels based on environmental and PGE specific data, and will be used by the developer to determine constraints which have been imposed by the PGS system and which may affect the normal processing of a PGE.

Tool Name: **Generate Product ID**

Description: This tool may be used to generate a unique product identifier. This identifier may be attached to file metadata to facilitate tracking of production output.

Inputs: Production run id

	Product id
	Science Software program id
Outputs:	Unique product identifier
	Status return value
Requirements:	PGSTK-1280
Synopsis:	
Notes:	If more than one product is being generated from the same PGE, then the appropriate product ID must be used as input to this routine when called from within the science software.

Tool Name:	**Get Configuration Data**
Description:	This tool may be used to import runtime configuration parameters into the PGE. Some of these configuration parameters may be specified by the PGE while others may be specified by the scheduling system.
Inputs:	none
Outputs:	Configuration parameters
	Status return value
Requirements:	PGSTK-1290
Synopsis:	
Notes:	(see requirement PGSTK-1290 in section 6 for examples of configuration parameters)

Tool Name:	**File Availability and Validity Query**
Description:	This tool may be used to extract information from the DAAC concerning the availability of PGE input files and the value of validity flags associated with the Q/A of input product data.
Inputs:	Query statement
Outputs:	Query results
	Status return value
Requirements:	PGSTK-1310
Synopsis:	

7.2.4 Memory Management Tools

Tool Name:	**Allocate Memory**

Description: This tool may be used to allocate dynamic memory for a new data structure. Pre-defined limits will be imposed on the amount of memory that may be allocated at any one time, or to any one process.

Inputs: Size of elemental data type

Number of elements for which memory is being requested

Outputs: Pointer to new memory buffer

Number of elements for which memory has been allocated

Status return value

Requirements: PGSTK-1240

Synopsis:

Notes: The tool will control the amount of memory that may be allocated at any one time.

Tool Name: **Re-allocate Memory**

Description: This tool may be used to reallocate dynamic memory in order to increase the size of an existing data structure. Pre-defined limits will be imposed on the amount of memory that may be reallocated at any one time, or to any one process.

Inputs: Pointer to original memory buffer

Size of elemental data type

Number of additional elements for which memory is being requested

Outputs: New Pointer to enlarged memory buffer

Number of additional elements for which memory has been allocated

Status return value

Requirements: PGSTK-1240

Synopsis:

Notes: The tool will control the amount of memory that may be allocated at any time.

Tool Name: **De-allocate Memory**

Description: This tool may be used to de-allocate memory that was previously allocated.

Inputs: Pointer to memory buffer

Outputs: none

Requirements: PGSTK-1240

Synopsis:

Notes: The tool can be used to free previously allocated memory that is no longer needed. These may be simple calls to standard C library functions such as malloc and free, but with checks on the actual machine memory available for (1). The usage of malloc, for example, may be limited.

7.2.5 Ancillary Data Access and Manipulation Tools

Ancillary data are defined in the ECS Functional Specification as any data other than Standard Data Products used in processing. This covers data from internal sources such as orbit ephemeris, spacecraft engineering data and calibration files and those from external sources such as meteorological data and DEMs. Standard products from other instruments strings may also be viewed as ancillary when used as support data. A further breakdown of relevance to understanding requirements is between essentially static data against dynamically changing data values. The requirements for and handling of ancillary data is viewed in 4 categories derived from this breakdown:

- Dynamically changing data from inside the EOSDIS [e.g. orbit ephemeris, instrument engineering data, products from other instruments]
- Static data from inside the EOSDIS [e.g. calibration files, look-up tables]
- Dynamically changing data from outside the EOSDIS [e.g. meteorological data from NOAA, radiances from GOES]
- Static data from outside the EOSDIS [e.g. DEMs, land-sea databases]

Access to the information will be in response to an algorithm request in the form of a pointer to a parameter in a data file. This pointer may take the form of a latitude and longitude or a similar two dimensional or three dimensional pointer. Time could also be a means of specifying the location of a required parameter value in the data set. An extension to this is a request for an area or the earth's surface, a volume of the atmosphere or a time period. This functionality might be used if efficient access to the data is required, or if differing interpolation methods are desired from those provided by the PGS Toolkit. For example, where repeated access to data from a small area is required by an algorithm, then receiving the whole area into memory from the ancillary file rather than calling the toolkit routine each time a value is required may dramatically reduce computer resource usage.

The functional requirements and data sets references in section 6.2.5. were derived from three sources; namely the Functional and Performance Requirements Specification for the EOSDIS Core System of 16th February 1993; secondly a requirements gathering exercise undertaken with instrument team representatives and finally the result of internal analysis.

A list of ancillary data files for which interfaces will be provided must be agreed and placed under configuration control as it is likely to change often as the algorithm designs evolve. The list should contain attributes associated with the data sets and their usage by the algorithms which allows effective modeling of the data ingest process and support of the toolkit design process. This configured list need not apply to static files from internal sources (e.g. algorithm calibration and control files) since they are associated with a single algorithm, are delivered to the DAAC with the algorithm and are configured as part of the algorithm. These latter types of ancillary files must be in one of the formats supported by the I/O functionality in the PGS toolkit.

The interfaces for ancillary data should be flexible to deal with additional data set requirements with minimal change. There will be two levels of interface functionality; firstly a set of low level functions providing functions such as open, close, input, output and simple search. These functions may be applied to most or all of the ancillary data sets required for access. The second level of functionality includes functions specific to particular data sets which require more complex manipulations. The number and extent of these more complex interfaces will only emerge as the algorithm designs evolve to an appropriate level of detail.

7.2.5.1 Dynamic ancillary data from internal sources.

Tool Name:	**Access Engineering and Ephemeris Data**
Description:	This tool(s) will provide interfaces to the 'internal' ancillary data for which the requirements in most cases will be the same for all algorithms; e.g., the requirements for orbit ephemeris are much the same in all PGS processing strings (see section 3.3.3 for an explanation of strings). The manipulations implied are relatively complex and need to be consistent throughout ECS. There may be several tools required having interfaces similar to that described below.
Requirements:	PGSTK-0730, 1251, 1252, 1253, 1254, 1255, 1256, 1257, 1258, 1259, 1260, 1261, 1262, 1263, 0730
Input:	type of ancillary data required; type of manipulation required (e.g. interpolation, extrapolation, averaging); location of required data; precision of data and calculations required (e.g. reconstituted orbit, full geolocation or time translation only).
Output:	value or values resulting from manipulation of data; error/status
Notes:	Specific requirements for manipulations and other aspects of this type of data are dealt with in greater depth elsewhere in this document (section 7.3.1.)
Synopsis:	

Tool Name:	**Access other EOS Products**
Description:	Products from EOS instruments used as ancillary data in algorithms related to other instruments also fall into this category and are assumed to be delivered in ECS standard format. Table C-6 of the Functional and Performance Requirements Specification implies both general and specific requirements for products to act as ancillary inputs to various strings (section 6.2.5.1.) This interface tool(s) will provide an interface to these products which will be the same for all strings. The use of standard ECS formats means that a small set of interface functions can deal with all possible formats (e.g. HDF raster or table). Generic I/O parameters are outlined below which should be similar for all of the necessary tools.
Requirements:	PGSTK-1251
Input:	ECS product identifier; ECS standard format type; location (in space and time) of required parameter value, parameters or values (i.e. multiple parameter and value extractions are permitted); type of manipulation required.
Output:	value of parameter or values of parameter or parameters; error/status.
Notes:	The most common manipulation is likely to be interpolation or extrapolation in space and/or time. The processing strings to which the ancillary data are input are all related to TRMM, EOS-AM1 spacecrafts or the AIRS instrument. Input ancillary data are provided by various instruments including those on these and additional spacecraft.
Synopsis:	

7.2.5.2 Static ancillary data from internal sources.

Tool Name:	**Access Algorithm Ancillary Data Files**
Description:	There is no specific level 3 requirement covering this category of data apart from statements about the input to the PGS of calibration coefficients. No further requirements from instrument teams sources were identified. However, this case is relatively simple and should be covered by general I/O tools provided by the PGS Toolkit. These data files are usually relatively small in volume in comparison with products and other types of ancillary data. There may be several associated with each algorithm. The number of these files will not be limited by a configured list since changes and additions are likely throughout algorithm development and updates. It will be necessary to limit the formats to ECS standard and ASCII only; i.e., to those formats which can be accessed using standard PGS toolkit I/O functions.
Requirements:	PGSTK-1265
Input:	data file name; location of required parameter value or values; format of required parameter.
Output:	value(s) of parameter(s);data file version number: error/status.
Notes:	A useful form for control and calibration files is the PARAMETER=VALUE format. If this special type of ASCII format is adopted by the algorithm developers, it may be efficient to extend the PGS Toolkit requirements to handle this format appropriately.
Synopsis:	

7.2.5.3 Static auxiliary data from external sources

Tool Name:	**Access data base of static geophysical data sets**
Description:	Data sets derived from external sources are often used by several algorithms and/or instruments. By grouping the requirements together, a general interface for static data set access can be proposed. The principal design elements of this interface must be to determine the formats of the data; i.e., raster, vector, table or something more complex. The integrity and qualities of the data must be considered when making this decision. In addition, PGS resource constraints and the characteristics of routines calling the interface(s) must be considered. For example, repeated access to a land-sea data set for sub-satellite point surface type definition requires that data be structured and accessed efficiently for single point extraction. Wider area extractions may require different interface characteristics and data set structure. The interface found below is generic. It may be necessary to develop several tools to cover these requirements.
Requirements:	PGSTK-0840, 0850, 0870, 0980, 1000, 1030, 1040, 1070
Input:	Data set name or names; location (in space) of required value or location or extremities of grid or volume to be extracted; manipulation required.
Output	Value of parameter at location or array(s) of values for grid or volume for each data set accessed;data file version; error/status.

Notes: No time input is necessary here. The output arrays would contain parameter values for the grid or volume requested which could be placed in memory for further rapid access or manipulations by the calling algorithm. It is recognized that this list of static data set requirements is likely to evolve as the algorithm design advances.

Synopsis:

7.2.5.4. Dynamic ancillary (external) data

Tool Name: **Access dynamically changing data from external sources**

Description: Dynamic data sets data require access by time as well as spatial location. The data sets below are complex in structure. Access to them will probably require a number of PGS tools. The critical issue here is the extent to which these data are prepared from their originating format and structure to be suitable for use in the algorithm calling the interface. It will be necessary for the incoming data to be unpacked from native format, restructured, re-ordered and output in ECS standard or other agreed intermediate format. The PGS toolkit routine will then access this intermediate format at the request of the algorithm requiring the data.

Requirements implied by Tables C-6 and E-1 of the Functional and Performance Requirements Specification (NAS5-60000 Attachment B) indicate a number of ancillary data sets. Most of these are correlative or for algorithm development or are required pre-launch only and therefore are not the concern of the PGS Toolkit. However, a number are ancillary data inputs to the PGS and are required post launch. Most of these are dynamic in nature (i.e. routinely updated) and these are listed as requirements here. These may require one tool per data set although several data sets may be accessible through one tool interface. The generic interface for all tools is found below.

Requirements: PGSTK-0921, & 0922, 0923, 0924, 0925, 0926, 0927, 0928, 0929, 0931, 1370.

Input: Data set name or names; location (time and space) of required value or location or extremities of grid or volume or set of time windows to be extracted; manipulation required (e.g.. interpolation).

Output Value of parameter at location or array(s) of values for grid, time windows or volume for each data set accessed; error/status

Synopsis:

Notes: The granularization of the incoming data is variable and the reorganization within the ECS is TBD. It is likely that similar parameters from one source will be grouped on a global or regional basis covering set time periods (e.g. 6 hours). The time location of the required value is then the key to the granule. It is recognized that this list of dynamic ancillary data set requirements is likely to evolve as the algorithm design advances.

7.2.6 Time and Date Conversion Tools

The ability to easily and accurately convert between different representations of time is crucial to EOS science data processing. The time and date conversion routines in the PGS Toolkit will convert between spacecraft time, ephemeris time, UTC and Julian date, as well as converting binary values to and from their corresponding ASCII format values. Time values are converted for use in science software and as parameters when performing geo-

coordinate transformations. In addition, converting time parameters to ASCII or to other more easily read formats facilitates the time values being added to metadata and to various processing logs in a human-readable form.

The spacecraft, UTC and ephemeris times and Julian dates used as input and output for these time and date conversion routines will be in Consultative Committee for Space Data Systems (CCSDS) standard time code formats. These formats are described in CCSDS 301.0-B-2, *Time Code Formats*, April 1990, Consultative Committee for Space Data Systems, Washington, DC. Specifically, EOS Spacecraft time codes are anticipated to be 64-bit values in the CCSDS day segmented (CDS) code format. This CDS time code is comprised of 16-bits representing the number of days since a reference epoch, 32-bits representing milliseconds of the day, and 16-bits representing microseconds, resulting in a resolution of approximately 15 μs. ASCII representations of time values will be in the CCSDS ASCII calendar segmented time code format.

Tool Name:	**Convert Time**
Description:	This tool converts binary time values in one system to binary values in three other time systems.
Requirement:	PGSTK-1170
Inputs:	Binary time parameter, fill values for the three parameters to be returned
	Desired ephemeris reference epoch (e.g. J1950, J1958, J2000)
	Desired Julian reference epoch (e.g. J1950, J1958, J2000)
Outputs:	Three converted time values, which will be leap second corrected as necessary
	Status return value
Synopsis:	

Tool Name:	**Convert Binary Time to ASCII Time**
Description:	This tool converts binary time values to ASCII time values.
Requirement:	PGSTK-1170
Input:	Binary time value
Output:	ASCII time value
	Status return value
Synopsis:	

Tool Name:	**Convert ASCII Time to Binary Time**
Description:	This tool converts ASCII time values to binary time values.
Requirement:	PGSTK-1170
Input:	ASCII time value

Output: Binary time value

Status return value

Synopsis:

Tool Name: **Convert Ephemeris Time Between Epochs**

Description: This Tool converts an ephemeris time in one reference epoch to an ephemeris time referenced to a different epoch.

Requirement: PGSTK-1215

Inputs: Ephemeris time parameter

Current ephemeris reference epoch (e.g. J1950, J1958, J2000)

Desired ephemeris reference epoch (e.g. J1950, J1958, J2000)

Output: Ephemeris time in new reference epoch

Status return value

Synopsis:

Tool Name: **Convert Julian Date Between Epochs**

Description: This Tool converts a Julian date value in one reference epoch to a Julian date value referenced to a different epoch.

Requirement: PGSTK-1220

Inputs: Julian date parameter

Current Julian reference epoch (e.g. J1950, J1958, J2000)

Desired Julian reference epoch (e.g. J1950, J1958, J2000)

Outputs: Julian date value in new reference epoch

Status return value

Synopsis:

Notes: Errors due to incorrect or out of range inputs or outputs will be trapped by these PGS Toolkit routines. Likewise, these routines will return an error status if any required conversion files or tables are missing or corrupted. Status conditions not related to errors will also be returned to provide confirmation of a successful time conversion.

Tool Name: **Convert to Foreign Platform Time**

Description: This Tool accesses the spacecraft time difference file for foreign platforms.

Requirement: PGSTK-1211

Inputs:	Time difference file name
	Local spacecraft time
Outputs:	Foreign spacecraft time
Synopsis:	
Notes:	The spacecraft time difference file is assumed to be provided externally.

7.3 PGS Toolkit Tools - Science

The tools described in this Section are designed to standardize certain science data and constants function calls. Their usage is encouraged for purpose of producing uniform products across the system and for reuse of products in subsequent processing. Usage will be assessed at integration time at the DAACs by a combination of automated code checkers and manual review.

7.3.1 Heavenly Body and Spacecraft Position Access Tools - Mandatory

These tools are required to provide commonalty and interoperability between data products of various EOS disciplines. Included here are heavenly body (Sun, Moon, major planets and bright stars), geolocation and spacecraft ephemeris access as well as coordinate system conversion routines. It is possible that many of the tools presented here will be heritage code or COTS software.

7.3.1.1 Spacecraft Ephemeris and Attitude Data Access Tools

Tool Name:	**Get Ephemeris and Attitude**
Description:	Provides access to spacecraft ephemeris and attitude data for a given time. The tool reads the ephemeris and attitude file, then interpolates the state vectors and spacecraft attitude to the specified time. If a range of times and a desired frequency of these times is specified as input, then vectors of corresponding positions, velocities and attitudes is returned. A flag that indicates reliability of the orbit and attitude data is also returned.
Requirements:	PGSTK-0720, 0740, 0710
Inputs:	Time or time range (UTC or spacecraft clock)
	State vector/attitude frequency (if range specified)
	Spacecraft identifier
Outputs:	Earth-Centered Inertial (ECI) position and velocity vector at the given time, or at specified time intervals within range
	Attitude at the given time, or at specified time intervals within range
	Orbit/attitude reliability flag, one for each time if multiple times requested
Synopsis:	
Notes:	The tool does not provide instrument pointing information since in most cases this will be a direct coordinate transformation from the spacecraft attitude; in cases where this is not true, the science software developers will be in the best position to provide the correction.

The specification for reliability of orbit and attitude is assumed to be provided by TBD.

It is assumed that instrument attitude with respect to the spacecraft is constant for a given launch; this assumption needs to be revalidated for platforms deployed subsequent to the EOS-AM platform.

7.3.1.2 Heavenly Body Position Access Tools

Tool Name:	**Get Earth to Planetary Body ECI Vector**
Description:	Computes the Earth-Centered Inertial (ECI) frame vector from the Earth to the Sun, Moon, or planets at a given time, or range of time(s).
Requirements:	PGSTK-0800, 0790
Inputs:	UTC time(s) and date(s)
	Planetary body flag, which indicates whether Sun, Moon, and/or indicated planet position(s) is desired
Outputs:	Earth-Centered Inertial (ECI) vector(s) from earth center to the Sun, Moon, or planet
Synopsis:	
Notes:	Sun, Moon and planetary ephemerides are assumed to be provided via standard astronomical tables.

Tool Name:	**Get Solar Time and Coordinates**
Description:	Computes local solar time, and right ascension and declination of the Sun, for a given standard time and position on the surface of the Earth.
Requirements:	PGSTK-0760, 0750
Inputs:	UTC time
	Position on the surface of the Earth (latitude or co-latitude, longitude)
Outputs:	Local solar time
	Solar right ascension and declination
Synopsis:	
Notes:	The declination is equivalent to the latitude of the sub-solar point; the difference between right ascension and longitude is Greenwich Mean Solar Time.
	For example, the solar position will be used to calculate the relationship between the solar illumination and the sensor viewing geometry.

Tool Name:	**Day/Night Indicator**
Description:	Determines whether a given point on the earth is in day or night, at a given time.

Requirements: PGSTK-0860, 0750, 0930

Inputs: UTC time and date

Position on the surface of the Earth (latitude or co-latitude, longitude)

Outputs: Flag to indicate whether point is in day or night

Synopsis:

Tool Name: **Get Greenwich Meridian ECI Coordinates**

Description: Computes the inertial position of the Greenwich meridian for a given time and position on the surface of the Earth.

Requirements: PGSTK-0900, 0750, 0930

Inputs: UTC time

Latitude or co-latitude and longitude

Outputs: Vectors with Earth-Centered Inertial (ECI) coordinates of the Greenwich meridian

Synopsis:

Tool Name: **Heavenly Body in Field-of-View Indicator**

Description: Given instrument parameters, returns a flag to indicate whether any of the major heavenly bodies (Sun, Moon, planets or stars with magnitudes greater than TBD) are in the instrument field-of-view.

Requirements: PGSTK-0780

Inputs: Instrument look vector

Instrument identifier

Outputs: Flag which indicates which one, if any, planetary body is in the instrument field-of-view

Synopsis:

Notes: Sun, Moon and planetary ephemerides, as well as major star locations, are assumed provided externally.

Half-angle of instrument field-of-view and instrument to platform transformation, as well as minimum star luminosity may be determined from a look-up table of static instrument data, assumed to exist.

Tool Name: **Get Bright Star Positions**

Description: Returns the position of all stars with luminosity greater than TBD.

Requirements: PGSTK-0820, 0790

Inputs: None

Outputs: Vectors of Earth-Centered Inertial (ECI) positions of stars

Vector of star identifiers

Notes: A table of major star locations are assumed provided externally.

Minimum star luminosity is TBD.

Synopsis:

Tool Name: **Get Greenwich Hour Angle**

Description: Returns the Greenwich Hour Angle, or hour angle of the Greenwich meridian in the Earth-Centered Inertial (ECI) frame, at a given time.

Requirements: PGSTK-0770

Inputs: UTC time

Outputs: Greenwich Hour Angle

Synopsis:

7.3.1.3 Coordinate System Conversion Tools

Tool Name: **Transform Between Reference Frames**

Description: Transforms between reference frames.

Requirements: PGSTK-1050, 1100, 0930

Inputs: Coordinates in original reference frame

Identifier of coordinate reference frame transformation to perform

+1, spacecraft reference to orbital reference

-1, orbital reference to spacecraft reference

+2, orbital reference to Earth reference ellipsoid

-2, Earth reference ellipsoid to orbital reference

+3, Earth reference ellipsoid to Space Oblique Mercator (SOM)

-3, SOM to Earth reference ellipsoid

+4, Earth-Centered Inertial (ECI) to Earth-Centered Rotating (ECR)

-4, ECR to ECI

+5, ECR to geodetic

-5, geodetic to ECR

+6, spacecraft reference to ECI

-6, ECI to spacecraft reference

Outputs: Coordinates in target reference frame

Synopsis:

Tool Name:	**Get Field-of-View Footprint**
Description:	Computes the projection of the instrument field-of-view on the earth.
Requirements:	PGSTK-1080, 0750, 0930
Inputs:	Instrument look vector Instrument identifier UTC time and date, or spacecraft time
Outputs:	Latitude (or co-latitude) and longitude of the near and far swath of the intersection of the instrument look vector with the earth's surface
Synopsis:	
Notes:	Half-angle of instrument field-of-view and instrument to platform transformation are determined from a look-up table of static instrument data, which is provided by TBD.

Tool Name:	**Get Sub-Satellite Point Position**
Description:	Returns the position and velocity vector of the sub-satellite point ("pierce point"), or nadir of the satellite on the earth's surface.
Requirements:	PGSTK-1060, 0950, 0750, 0930
Inputs:	UTC time and date, or spacecraft clock time
Outputs:	Latitude or co-latitude, longitude, and velocity vector of sub-satellite point
Synopsis:	
Notes:	Spacecraft ephemeris access is required for this tool.

7.3.1.4 Geo-Coordinate Transformation Tools

Tool Name:	**Map Projection**
Description:	These tools are required to provide the transformation from the Cartesian reference ellipsoid frame coordinates provided by the geolocation functions to some pre-defined geographic plane coordinate systems (map projection). Four projections have been identified as requirements: Space Oblique Mercator, Universal Transverse Mercator, Polar Stereographic, Goodes Interrupted Homolosine Projection. To allow for a large number of values to be transformed quickly, these tools will accept arrays of input vectors (x, y) for transformation. It is assumed that the algorithm will provide an array of vectors to the tool in a single call and will receive the transformed vectors on return.

The transformations for each projection will be bi-directional. A direct bi-directional conversion from the Space Oblique Mercator to the Universal Transverse Mercator will be provided.

Requirements: PGSTK-1500,1501,1502

Input: number of coordinate values to be transformed, array of coordinates in the original coordinate system, original coordinate system code, transformed coordinate system code

Output: array of transformed coordinates, error/status

Synopsis:

Notes: The coordinate transformation will be based on the map projections specified by the USGS. t is recognized that the requirements for map projections may evolve as the algorithm design advances.

7.3.2 Math & Statistical Support Tools - Optional

This section describes requirements for basic math and statistics functions to be supplied by COTS software. The library of functions must be maintainable as the PGS evolves so that new functionality does not make old software obsolete. Interfaces with math packages must be strictly controlled to protect users from version changes.

Tool Name: **Math and Statistics**

Description: A suite of standard mathematical manipulation functions is provided in a single uniform package.

Requirements: PGSTK-1245

Inputs:

a. parameters of equation, input matrix and desired manipulation

b. input data and output grid

c. function and limits

d. equation and coefficients

e. function

f. data and statistics desired

g. function

Outputs:

a. solutions of equation, output matrix or result

b. data on desired grid

c. integral of function

d. roots of equation

e. min/max

f. output statistics

g. transform

Synopsis:

Notes: User feedback will determine which math tools will be included in the Toolkit. The interfaces to these tools will remain fixed once they are implemented.

7.3.3 Constants and Unit Conversions - Optional

7.3.3.1 Math, Physical and Instrument Constants

The use of constant values for parameters within ECS is important for consistency between algorithms and products. A set of standard math and physical parameters (e.g. pi, speed of light etc.) and their values, proposed by the Data Processing Focus Team will be accessible from the PGS Toolkit. The tool will provide the functionality to include the approved parameter values in the algorithms. A similar interface will be provided for instrument specific constants, to provide consistency between different algorithms using data from the same instrument.

Tool Name: **Get Constant Value**

Description: This tool will return a constant value in the form of: parameter=value

Requirements: PGSTK- 1520,1521,1522

Input: reserved keyword (e.g. $pi)

Output: value for parameter approved by DPFT, IWG or other cognizant body

Synopsis:

7.3.3.2 Units Conversions

The PGS toolkit will provide functions to perform common units conversions or parameter translations in a consistent manner. The preliminary list of conversions will include: length conversions (e.g. feet to m, etc.), area conversions, volume conversions, temperature conversions, Planck's function.

Tool Name: **Convert Units**

Description: The functions in this group will be restricted to conversions that can be achieved with standard formulae, without the need for additional external data.

Requirements: PGSTK-1530,1531

Input: parameter keyword, existing units, required units, value(s)

Output: converted value, error/status

Synopsis:

Notes: It is premature to establish the list of conversions as the algorithm development has not reached an appropriate level of detail. A basic set of conversions common in earth

observation data processing schemes will be produced for the first release of this functionality (see description above). For subsequent releases of the toolkit, new requirements which emerge from the evolving algorithm designs will be assessed for inclusion in the PGS toolkit requirements.

7.3.4 Graphics Support Tools - Optional

These tools will support graphics file output from science production processes. It is anticipated that the primary usage of these tools in the PGS will be for analyzing Q/A data. A flag can be set which will enable delivery of these files to the SCF. These tools will also be used to process HDF files output from production processes, although the specific graphics file format which will be supported is TBD. These tools may be a subset of IMS users support software.

Tool Name:	**Write Graphics File**
Description:	Writes a file of output in graphical format, suitable for ingest into a visualization package.
Requirements:	PGSTK-1410
Inputs:	Data to write
Outputs:	none
Synopsis:	
Notes:	Graphical format(s) are TBD.

Tool Name:	**Do Image Processing**
Description:	Performs image processing to a block of data
Requirements:	PGSTK-1415
Inputs:	Data to process
	Type of image processing to apply
Outputs:	Statistics on data: correlations, regressions, variances, etc.
	Filtered/enhanced image
Synopsis:	
Notes:	It is anticipated that many different tools will be delivered to meet this requirement.

8. Proposed PGS Toolkit Delivery Priorities

The PGS Toolkit is to be delivered in stages. Due to the lateness of Contract award, the accelerated launch schedule and the inclusion of TRMM instrument support, most software required in this Specification should be delivered within 18 months of Contract award. The most important software should be delivered within one year of Contract award. It is expected that initial software implementations will be upgraded or enhanced on a regular basis through consultation with algorithm support teams, scientists, developers and Hughes team systems engineers. Extensive user interaction and comment is recommended throughout the delivery process. The implementation schedule will be a compromise between what science software developers require and what is feasible to deliver.

We propose the following priorities for deliveries to the SCFs:

Highest Priority

Preliminary list of routines, calling sequences and description

Interface Control Document

Preliminary Users Guide

On-line documentation of Toolkit routines for science software development

Preliminary PGS Scheduling Emulation Tools

File I/O Tools (Preliminary L0-L4, metadata and browse)

Time/Date Format Conversion Tools

COTS Math/Stat Libraries

Geo-location Tools

Geo-coordinate Transformation Tools

Lower Priority

Error and Status Reporting Tools

Operation Support Tools

Physical Constants

Ancillary/Auxiliary Data Access Tools

PGS Scheduling Emulation Tools

Simulated L0 Data

Simulated Orbit Data

Lowest Priority

Graphics Support Tools

Appendix A. Requirements Traceability Matrix

In this Appendix, we present a trace of the requirements listed in this document to their sources.

In Table A-1:

"Section" and "Rqt #" refer to Sections 5, 6, and 7 of this document.

"L3 Rqt #" refers to the parent requirement from NAS5-60000, Attachment B, "Functional and Performance Requirements Specification for the EOSDIS Core System" [Level 3 requirements], dated 16 February 1993. In this column, "PGS-apdx" refers to Appendix C of NAS5-60000. "PGS-9999" refers to requirements for which no Level 3 requirement applies.

"Doc #" refers to the source document. See Table 2 of this appendix.

"Instr/Org" refers to the instrument team or organization that initiated the requirement. In this column, "JPL" refers to all JPL-based instrument teams, including AIRS, ASTER, MISR, SeaWinds and NSCAT; "Derived" means the requirement was derived from Level 3 requirements by the ECS contractor.

"Trace #" refers to the internal trace identification, assigned to all explicit potential requirements received from the science software teams.

The table is sorted on requirement number. Duplicate PGSTK numbers correspond to the case where several requirements were combined into one.

Table A-1. Requirements Traceability Matrix (1 of 4)

Section	Rqt #	L3 Rqt #	Doc #	Instr/Org	Trace #
5.2.1	PGSTK-0010	PGS-1030		Derived	
5.2.1	PGSTK-0020	PGS-1030		Derived	
5.2.1	PGSTK-0040	PGS-1030		Derived	
5.2.2	PGSTK-0050	PGS-1030	93-067	DPFT III	PGS-TR-073
5.2.2	PGSTK-0060	PGS-1030	93-068	DPFT II	PGS-TR-080
5.2.2	PGSTK-0080	PGS-1030	93-073	CERES	PGS-TR-116
5.2.2	PGSTK-0090	PGS-1030		Derived	
5.2.3	PGSTK-0100	PGS-1030		Derived	
6.2	PGSTK-0120	PGS-0602		Derived	
6.2	PGSTK-0121	PGS-0602		Derived	
5.2.3	PGSTK-0123	PGS-1030	93-066	JPL	PGS-TR-057
5.2.4	PGSTK-0140	PGS-1030	93-023	JPL	PGS-TR-085
5.2.5	PGSTK-0160	PGS-1030		Derived	
5.2.5	PGSTK-0170	PGS-1030		Derived	
6.2.0	PGSTK-0180	PGS-1000		Derived	
6.2.1.1	PGSTK-0190	PGS-0435		Derived	
6.2.1.1	PGSTK-0200	PGS-0435		Derived	

Table A-1. Requirements Traceability Matrix (2 of 4)

6.2.1.1	PGSTK-0210	PGS-0435		Derived	
6.2.1.1	PGSTK-0220	PGS-0435	93-067	AIRS	PGS-TR-078
6.2.1.1	PGSTK-0230	PGS-0435	93-076	MODIS	PGS-TR-167
6.2.1.1	PGSTK-0240	PGS-0435	93-056	LIS	PGS-TR-011
6.2.1.2	PGSTK-0250	PGS-0970	93-077	MODIS	PGS-TR-153
6.2.1.2	PGSTK-0270	PGS-0970	93-078	MOPITT	PGS-TR-174
6.2.1.2	PGSTK-0271	PGS-0970		Derived	
6.2.1.2	PGSTK-0290	PGS-0970		Derived	
6.2.1.2	PGSTK-0300	PGS-0970		Derived	
6.2.1.2	PGSTK-0310	PGS-0410		Derived	
6.2.1.2	PGSTK-0320	PGS-0970	93-076	MODIS	PGS-TR-166
6.2.1.3	PGSTK-0360	PGS-1260	93-073	CERES	PGS-TR-124
6.2.1.4	PGSTK-0370	PGS-0510		Derived	
6.2.1.4	PGSTK-0380	PGS-0510		Derived	
6.2.1.4	PGSTK-0380	PGS-0510	93-065	AIRS	PGS-TR-035
6.2.1.4	PGSTK-0390	PGS-0510		Derived	
6.2.1.4	PGSTK-0400	PGS-0510		Derived	
6.2.1.4	PGSTK-0410	PGS-0510		Derived	
6.2.1.4	PGSTK-0430	PGS-0510		Derived	
6.2.1.4	PGSTK-0440	PGS-0510		Derived	
6.2.1.4	PGSTK-0450	PGS-0510	93-064	Derived	
6.2.1.5	PGSTK-0490	PGS-1050		Derived	
6.2.1.5	PGSTK-0500	PGS-1050	93-068	DPFT II	PGS-TR-082
6.2.1.5	PGSTK-0510	PGS-1100	93-073	MODIS	PGS-TR-141
6.2.1.6	PGSTK-0520	PGS-1010		Derived	
6.2.1.6	PGSTK-0530	PGS-1010		Derived	
6.2.1.6	PGSTK-0531	PGS-1010		Derived	
6.2.1.6	PGSTK-0532	PGS-1010		Derived	
6.2.1.6	PGSTK-0533	PGS-1010		Derived	
6.2.1.6	PGSTK-0534	PGS-1010		Derived	
6.2.1.6	PGSTK-0540	PGS-1010		Derived	
6.2.1.7	PGSTK-0550	PGS-0970	93-064	ASTER	PGS-TR-025
6.2.1.7	PGSTK-0570	PGS-0530		Derived	
6.2.2	PGSTK-0580	PGS-0990		Derived	
6.2.2	PGSTK-0590	PGS-1000		Derived	
6.2.2	PGSTK-0600	PGS-0990		Derived	
6.2.2	PGSTK-0610	PGS-1000		Derived	
6.2.2	PGSTK-0620	PGS-1000	93-067	DPFT III	PGS-TR-074
6.2.2	PGSTK-0630	PGS-1000	93-064	ASTER	PGS-TR-028
6.2.2	PGSTK-0650	PGS-1000	93-065	AIRS	PGS-TR-039

.c. Table A-1. Requirements Traceability Matrix (3 of 4)

6.3.1	PGSTK-0680	PGS-1015	93-073	CERES	PGS-TR-118
6.3.1.1	PGSTK-0710	PGS-1015	93-066	JPL	PGS-TR-044
6.3.1.1	PGSTK-0720	PGS-1015		Derived	
6.2.5.1	PGSTK-0730	PGS-1015		Derived	
6.3.1.1	PGSTK-0740	PGS-1015	93-079	JPL	PGS-TR-156
6.3.1	PGSTK-0750	PGS-1015	93-073	CERES	PGS-TR-131
6.3.1.2	PGSTK-0760	PGS-1015		Derived	
6.3.1.2	PGSTK-0770	PGS-1015		Derived	
6.3.1.2	PGSTK-0780	PGS-1015	93-066	JPL	PGS-TR-046
6.3.1.2	PGSTK-0790	PGS-1015	93-066	JPL	PGS-TR-047
6.3.1.2	PGSTK-0800	PGS-1015	93-073	MODIS	PGS-TR-132
6.3.1.2	PGSTK-0820	PGS-1015	93-073	MODIS	PGS-TR-134
6.2.5.3	PGSTK-0840	PGS-0490		Derived	
6.2.5.3	PGSTK-0850	PGS-0490		Derived	
6.3.1.2	PGSTK-0860	PGS-1015	93-056	LIS	PGS-TR-006
6.2.5.3	PGSTK-0870	PGS-0490	93-056	LIS	PGS-TR-007
6.2.5.3	PGSTK-0870	PGS-1015	93-023	JPL	PGS-TR-090
6.3.1.2	PGSTK-0900	PGS-1015	93-073	Gelocation	PGS-TR-106
6.2.5.4	PGSTK-0921	PGS-0520		Derived	
6.2.5.4	PGSTK-0922	PGS-0520		Derived	
6.2.5.4	PGSTK-0923	PGS-0520		Derived	
6.2.5.4	PGSTK-0924	PGS-0520		Derived	
6.2.5.1	PGSTK-0925	PGS-1015		Derived	
6.2.5.4	PGSTK-0926	PGS-0520		Derived	
6.2.5.4	PGSTK-0927	PGS-0520		Derived	
6.2.5.4	PGSTK-0928	PGS-0520		Derived	
6.2.5.4	PGSTK-0929	PGS-0520		Derived	
6.3.1	PGSTK-0930	PGS-1015	93-073	MODIS	PGS-TR-135
6.2.5.4	PGSTK-0931	PGS-0520		Derived	
6.2.5.3	PGSTK-0980	PGS-0490		Derived	
6.2.5.3	PGSTK-0980	PGS-0490	93-065	AIRS	PGS-TR-038
6.2.5.3	PGSTK-0980	PGS-0490	93-073	Gelocation	PGS-TR-109
6.2.5.3	PGSTK-1000	PGS-0490		Derived	
6.2.5.3	PGSTK-1030	PGS-0490	93-066	JPL	PGS-TR-050
6.2.5.3	PGSTK-1040	PGS-0490	93-073	MODIS	PGS-TR-136
6.3.1.3	PGSTK-1050	PGS-1015		Derived	
6.2.3.5	PGSTK-1050	PGS-1015	93-063	LIS	PGS-TR-017
6.3.1.3	PGSTK-1050	PGS-1015	93-073	Gelocation	PGS-TR-108
6.3.1.3	PGSTK-1060	PGS-1015		Derived	

6.2.6	PGSTK-1170	PGS-0435	93-063	LIS	PGS-TR-019
6.2.6	PGSTK-1180	PGS-0435	93-063	LIS	PGS-TR-020
6.2.6	PGSTK-1190	PGS-0435	93-063	LIS	PGS-TR-021
6.2.6	PGSTK-1195	PGS-0435		Derived	
6.2.6	PGSTK-1200	PGS-0435	93-073	CERES	PGS-TR-119
6.2.6	PGSTK-1200	PGS-1015	93-066	JPL	PGS-TR-052
6.2.6	PGSTK-1210	PGS-0435	93-073	CERES	PGS-TR-120
6.2.6	PGSTK-1211	PGS-9999		Derived	
6.2.6	PGSTK-1215	PGS-0435		Derived	
6.2.6	PGSTK-1220	PGS-0435	93-079	JPL	PGS-TR-154
6.2.4.1	PGSTK-1230	PGS-9999		Derived	
6.2.4.2	PGSTK-1240	PGS-9999		Derived	
6.3.2	PGSTK-1245	PGS-1020	93-067	DPFT III	PGS-TR-053
6.2.5.1	PGSTK-1251	PGS-1015		Derived	
6.2.5.1	PGSTK-1252	PGS-apdx		Derived	
6.2.5.1	PGSTK-1253	PGS-apdx		Derived	
6.2.5.1	PGSTK-1254	PGS-apdx		Derived	
6.2.5.1	PGSTK-1255	PGS-apdx		Derived	
6.2.5.1	PGSTK-1256	PGS-apdx		Derived	
6.2.5.1	PGSTK-1257	PGS-apdx		Derived	
6.2.5.1	PGSTK-1258	PGS-apdx		Derived	
6.2.5.1	PGSTK-1259	PGS-apdx		Derived	
6.2.5.1	PGSTK-1260	PGS-apdx		Derived	
6.2.5.1	PGSTK-1261	PGS-apdx		Derived	
6.2.5.1	PGSTK-1262	PGS-apdx		Derived	
6.2.5.1	PGSTK-1263	PGS-apdx		Derived	
6.2.5.2	PGSTK-1265	PGS-1015		Derived	
6.2.5.2	PGSTK-1265	PGS-1015	93-073	Gelocation	PGS-TR-105
6.2.3	PGSTK-1280	PGS-0510		Derived	
6.2.3	PGSTK-1290	PGS-0980	93-073	CERES	PGS-TR-128
6.2.3	PGSTK-1310	PGS-0980	93-077	MODIS	PGS-TR-150
6.2.5	PGSTK-1360	PGS-1015		Derived	
6.2.5	PGSTK-1361	PGS-0510		Derived	
6.2.5.4	PGSTK-1370	PGS-1015	93-066	JPL	PGS-TR-056
6.3.4	PGSTK-1410	PGS-1025	93-067	DPFT III	PGS-TR-079
6.3.4	PGSTK-1415	PGS-1025	93-023	JPL	PGS-TR-092
6.3.4	PGSTK-1415	PGS-1025	93-023	JPL	PGS-TR-098
6.3.1.4	PGSTK-1500	PGS-1015		Derived	
6.3.1.4	PGSTK-1501	PGS-1015	93-063	LIS	PGS-TR-018
6.3.1.4	PGSTK-1502	PGS-1015		Derived	
6.3.3.1	PGSTK-1520	PGS-1025		Derived	
6.3.3.1	PGSTK-1521	PGS-0458		Derived	
6.3.3.1	PGSTK-1522	PGS-1020		Derived	
6.3.3.2	PGSTK-1530	PGS-1020		Derived	
6.3.3.2	PGSTK-1530	PGS-1020	93-056	LIS	PGS-TR-005
6.3.3.2	PGSTK-1530	PGS-1020	93-023	JPL	PGS-TR-097
6.3.3.2	PGSTK-1531	PGS-1020		Derived	

.c. Table A-2 Source Documents

Doc Number	Name
93-023	Response to NASA PGS Toolkit Study, JPL (MISR/AIRS/ASTER), 11/92
93-055	Preliminary Draft: ECS PGS Toolkit Requirement Specification, 6/29/93
93-056	Comments on Prelim Draft PGS Toolkit Rqts Spec, LIS Team , 7/6/93
93-063	Private Communication: Steve Goodman/LIS, 7/16/93
93-064	ASTER Product Generation System Concept, 6/10/93
93-065	PGS Operations Concept for the AIRS Instrument, V1.0.2, 5/7/93
93-066	Response to Preliminary Draft of PGS Toolkit Rqt Spec by JPL, 7/13/93
93-067	DPFT III notes - Hughes Internal, 6/23/93
93-068	Minutes: 2nd Data Processing Focus Team mtg (DPFT II), 3/11/93
93-073	Hughes Trip Notes:
	CERES 7/22/93
	Geolocation Meeting - Hughes Internal 6/9/93
	MODIS 6/8/93
	AIRS 5/25/93
	DPFT III NASA/GSFC (T.Meyer, Coord.) 6/23/93
	MODIS Level 1A SRR (MODIS SDST) 5/11/93
93-074	Geolocation Meeting Slides and Handouts - 6/9/93
93-076	MODIS Comments on Prelim Draft PGS Toolkit Rqts Spec, 7/26/93
93-077	MODIS Data Processing Operations Concept Draft, 7/13/93
93-078	S.Scott (NASA/GSFC) private communication, MOPITT trip notes, 7/15/93
	MOPITT comments on Prelim Draft PGS Toolkit Rqts Spec 7/15/93
93-079	Comments on Prelim Draft PGS Toolkit Rqts Spec, J.Brown, SeaWinds/JPL, 7/13/93

This page intentionally left blank.

Appendix B. POSIX System Calls Usage Policy

This appendix outlines the usage policy for the set of POSIX system API calls as outlined in:

IEEE Std 1003.1: *POSIX Part 1: System Application Program Interface (API) [C Language]*

IEEE Std 1003.9: *POSIX FORTRAN77 Language Interfaces, Part 1: Binding for System Application Program Interface [API]*

In general, the policy attempts to guard access to routines that may impact the PGS Scheduler's responsibility for system resource management. This will be accomplished by restricting access to the standard POSIX system calls, as described below. The complete set of routines is listed in the "Identifier Index" of IEEE Std 1003.1, and in the body of IEEE Std 1003.9.

Table B-1 provides general policy "guidelines" for various classes of system routines. These guidelines are then used in determining the appropriate disposition for each of the POSIX system call functions on a one-by-one basis. The general policy guidelines include:

Toolkit	Access to the described functionality is provided via a "shadowing" routine in the PGS Toolkit. The Toolkit routine may be a simple subroutine call (or macro) wrapper around the "shadowed" function, or it may provide additional functionality that may be useful to the PGS Scheduler in accomplishing its resource management objectives. Direct calls to the respective POSIX API calls are prohibited within science algorithm code.
Prohibited	Access to the described functionality is prohibited. Direct calls to the respective POSIX API calls are prohibited within science algorithm code.
Allowed	Access to the described functionality is allowed through the standard POSIX API calls.

The algorithm integration and test facility will include "code checkers" to screen science algorithms for adherence. These code checkers will be provided as part of the PGS Toolkit to support the development of policy compliant algorithms. This should greatly facilitate the algorithm integration and test procedure.

Table B-1. POSIX Call Guidelines By Class.

Class	Description	Policy Guideline
Process control	Process creation and termination; interprocess signaling and synchronization	Toolkit
Memory	Memory allocation, deallocation, and mapping	Toolkit
File I/O	File I/O routines; directory manipulation routines	Toolkit
Stream I/O	Stream I/O routines	Toolkit
Error / environment	Error recording and reporting; environment access	Toolkit
Ownership	Process ownership and groups; file ownership, permissions, and creation/access times	Prohibited
Miscellaneous	Math, "is...", "str...", and time functions	Allowed
Terminal I/O	Terminal I/O and characteristics	Allowed
Status	System and resource status (read only)	Allowed

Tables B-2 through B-10 constitute a listing of the entire set of POSIX C API calls, organized by class and policy as described above. Table B-11 provides a listing of the FORTRAN77 specific language library calls which do not have C API counterparts.

Table B-2. POSIX Calls: Process Control.

Toolkit routines	Prohibited routines	Allowed routines
abort()		setjmp()
alarm(), PXFALARM()		longjmp()
exec...(), PXFEXEC...()		
exit(), PXFEXIT()		
_exit(), PXFFASTEXIT()		
fork(), PXFFORK()		
kill(), PXFKILL()		
pause(), PXFPAUSE()		
sig...(), PXFSIG...()		
sleep(), PXFSLEEP()		
wait(), PXFWAIT()		
waitpid(), PXFWAITPID()		

Table B-3. POSIX Calls: Memory.

Toolkit routines	Prohibited routines	Allowed routines
calloc()		
free()		
malloc()		
realloc()		

Table B-4. POSIX Calls: File I/O.

Toolkit routines	Prohibited routines	Allowed routines
access(), PXFACCESS()	chdir(), PXFCHDIR()	
close(), PXFCLOSE()	closedir(), PXFCLOSEDIR()	
creat(), PXFCREAT()	fpathconf(), PXFFPATHCONF()	
dup(), PXFDUP()	getcwd(), PXFGETCWD()	
dup2(), PXFDUP2()	link(), PXFLINK()	
lseek(), PXFLSEEK()	mkdir(), PXFMKDIR()	
open(), PXFOPEN()	mkfifo(), PXFMKFIFO()	
pipe(), PXFPIPE()	opendir(), PXFOPENDIR()	
read(), PXFREAD()	pathconf(), PXFPATHCONF()	
remove()	readdir(), PXFREADDIR()	
rename(), PXFRENAME()	rewinddir(), PXFREWINDDIR()	
rewind()	rmdir(), PXFRMDIR()	
tmpfile()	setbuf()	
tmpnam()		
unlink(), PXFUNLINK()		
write(), PXFWRITE()		

Table B-5. POSIX Calls: Stream I/O.

Toolkit routines	Prohibited routines	Allowed routines
clearerr()		
fclose()		
fcntl(), PXFFCNTL()		
fdopen()		
feof()		
ferror()		
fflush()		
fgetc(), PXFFGETC()		
fgets()		
fileno()		
fopen()		
fprintf()		
fputc(), PXFFPUTC()		
fputs()		
fread()		
freopen()		
fscanf()		
fseek(), PXFFSEEK()		
ftell(), PXFFTELL()		
fwrite()		
PXFFDOPEN()		
PXFFILENO()		
PXFFLUSH()		
PXFPOSIXIO()		

Table B-6. POSIX Calls: Error/environment.

Toolkit routines	Prohibited routines	Allowed routines
		assert() getenv(), PXFGETENV() perror() IPXFARGC() PXCLEARENV() PXFGETARG() PXSETENV()

Table B-7. POSIX Calls: Ownership.

Toolkit routines	Prohibited routines	Allowed routines
getpid(), PXFGETPID() getppid(), PXFGETPPID()	chmod(), PXFCHMOD() chown(), PXFCHOWN() getegid(), PXFGETEGID() geteuid(), PXFGETEUID() getgid(), PXFGETGID() getgrgid(), PXFGETGRGID() getgrnam(), PXFGETGRNAM() getgroups(), PXFGETGROUPS() getlogin(), PXFGETLOGIN() getpgrp(), PXFGETPGRP() getpwnam(), PXFGETPWNAM() getpwuid(), PXFGETPWUID() getuid(), PXFGETUID() setgid(), PXFSETGID() setpgid(), PXFSETPGID() setsid(), PXFSETSID() setuid(), PXFSETUID() umask(), PXFUMASK() utime(), PXFUTIME()	

Table B-8. POSIX Calls: Miscellaneous.

Toolkit routines	Prohibited routines	Allowed routines
		abs()
		acos()
		asctime()
		asin()
		atan()
		atan2()
		atof()
		atoi()
		atol()
		bsearch()
		ceil()
		cos()
		cosh()
		ctime()
		exp()
		fabs()
		floor()
		fmod()
		frexp()
		gmtime()
		is...()
		localtime(), PXFLOCALTIME()
		log()
		log10()
		mktime()
		modf()
		pow()
		qsort()
		rand()
		setlocale()
		sin()
		sinh()
		sqrt()
		srand()
		tan()
		tanh()
		time(), PXFTIME()
		tolower()
		toupper()
		tzset()
		IAND()
		IOR()
		NOT()
		IPXFLENTRIM()
		PXFUCOMPARE()

Table B-9. POSIX Calls: Terminal I/O.

Toolkit routines	Prohibited routines	Allowed routines
		cfgetispeed(), PXF...() cfgetospeed(), PXF...() cfsetispeed(), PXF...() cfsetospeed(), PXF...() ctermid(), PXFCTERMID() getc(), PXFGETC() getchar() gets() isatty(), PXFISATTY() printf() putc(), PXFPUTC() putchar() puts() scanf() sprintf() sscanf() tc...(), PXFTC...() ttyname(), PXFTTYNAME() ungetc()

Table B-10. POSIX Calls: Status.

Toolkit routines	Prohibited routines	Allowed routines
fstat(), PXFFSTAT() stat(), PXFSTAT() PXFIS...()		sysconf(), PXFSYSCONF() times(), PXFTIMES() uname(), PXFUNAME()

Table B-11. POSIX Calls: FORTRAN77 Language Library

Toolkit routines	Prohibited routines	Allowed routines
		PXFCALLSUBHANDLE() IPXFCONST() PXFCONST() PXFISCONST() PXFSTRUCTCOPY() PXFSTRUCTCREATE() PXFSTRUCTFREE() PXFSUBHANDLE()PXF<TYPE>GET() PXF<TYPE>SET() PXFA<TYPE>GET() PXFA<TYPE>SET() PXFE<TYPE>GET() PXFE<TYPE>SET()

Appendix C. Open Issues

In this section we list issues which have not been resolved at the time of the present document release, but which are potential requirements to be included in a later version.

The number listed at the end of each requirement is an internal identifier for tracking requirements given collected from the science software teams; these PGS-TR numbers differ from the PGSTK numbers that appear in section 6, which represent firm Toolkit requirements. Also listed is the science team or organization from whom the requirement was received.

PGS Scheduler Emulation at the SCF

Potential Requirements:

The PGS Toolkit shall contain tools to dynamically schedule scripts which emulate data flow through the processing string, and shall be able to handle conditional processing scenarios. The usage of such tools at the SCF should minimally affect integration of science software at the DAAC. (PGS-TR-170, MOPITT)

Provide a functional copy of the PGS environment, including scheduler, task initiator, etc. as part of the SCF toolkit. (PGS-TR-164, MODIS)

Simulate the scheduling system at the SCF. (PGS-TR-072, DPFT III)

All requirements for data to kick off in the production environment must be testable at the SCF. (PGS-TR-117, CERES)

Spacecraft Environment Simulation

Potential Requirements:

The PGS Toolkit shall contain tools which simulate the spacecraft environment, spacecraft characteristics, orbit parameters, and instrument characteristics. Provision shall be made for introduction of anomalies for testing Toolkit functions and interfaces. (PGS-TR-171, MOPITT)

The PGS Toolkit shall provide a self-consistent universe simulator for orbital data which contains data on angles, earth location, etc. for a 2 week period in summer and another 2 week period in winter. (PGS-TR-147, MOPITT)

Provide access to the GSFC/FDF EPHEM software. (PGS-TR-173, LIS)

The SCF Toolkit shall provide access to simulated orbit data for at least 1 day and 1 night (15 consecutive orbits). (PGS-TR-103, Geolocation mtg)

Standards Compliance

Potential Requirements:

The PGS Toolkit shall be XPG compliant.

Note:

This issue is currently being investigated, in particular its implications with regard to high-performance computing.

Ancillary /Auxiliary Data

Potential Requirements:

Provide access to startracker data and history. (PGS-TR-043, JPL)

Provide tools to access NOAA Radiosonde data, TIGR data, Global Circulation Models, spectral line transmission data. (PGS-TR-037, AIRS)

Provide access to tidal information. (PGS-TR-045, JPL)

Provide information about the dynamic extent of sea ice. (PGS-TR-048, JPL)

Provide access to a global model of the tropopause. (PGS-TR-016, LIS)

Translate NMC files into indexed HDF format. (PGS-TR-145, MOPITT)

The PGS Toolkit shall contain tools to provide access to land/snow and land/ice data. (PGS-TR-121, CERES)

The PGS Toolkit shall contain tools to provide access to snow cover and earth albedo data sets, which are updated daily. (PGS-TR-127, CERES)

The PGS Toolkit shall contain a tool that provides access to a ground control point chip library. (PGS-TR-137, MODIS)

Provide tools to access reference images, e.g., chip files. (PGS-TR-091, JPL)

The PGS Toolkit shall contain tools to provide access to the extent of sea-ice at better than 1 week intervals. (PGS-TR-157, JPL)

Provide a tool to determine if a given latitude (+/-dd.d) longitude (ddd.d), one tenth degree resolution, is land/coastline, 95% sea/ice covered sea. (PGS-TR-014, LIS)

Note:

A complete list of auxiliary data which the Toolkit will support and when that support will occur is TBD.

Miscellaneous Geolocation Tools

Potential Requirement:

Provide a tool to return UTC time interval, start and duration that a given earth station is within the field of view of a cone of sight vectors given ephemeris times and position/velocity state vectors. (PGS-TR-015, LIS)

The PGS Toolkit shall contain tools to determine the fraction of instrument field-of-view occupied by a given geographical area. (PGS-TR-052-JPL)

Languages

Potential Requirement:

The PGS Toolkit shall support bindings to FORTRAN90.

Heap Manager

Potential Requirement:

Provide a heap manager for memory allocation and garbage collection. (PGS-TR-094, JPL)

Script Language

Potential Requirement:

Provide any scripting language tools in the Bourne shell, as that is the easiest Unix shell syntax to use. (PGS-TR-001, LIS)

Graphics

Potential Requirements:

Provide tools to facilitate the formation of mosaics and tiling of images. (PGS-TR-104, Geolocation mtg)

Provide a list of officially supported graphics formats for graphical file output. (PGS-TR-162, MODIS)

I/O Word Alignment and Byte Ordering

Potential Requirement:

The PGS Toolkit shall provide tools to ensure that non-HDF format data will be easily portable, including (1) a tool to make ordering of bytes transparent, irrespective of what hardware the algorithm runs on and (2) tools to eliminate buffering due to word alignment requirements on different machines.

Notes:

These tools would be used to correct for the byte ordering conflict that may arise when attempting to read data from a file that was created on a different, and incompatible, computing platforms. The order in which machines write bytes for, say, a 2-byte integer, varies on different machines. This implies that a structure consisting of 5 bytes, will be fit to 3 words, and actually be 6 bytes long-one byte being padding. These tools would ensure that all machines will be able to read one another's files into the proper structures without concern for padding considerations.

Computing Platforms

Potential Requirement:

The SCF Toolkit shall contain a version that runs on Cray C90 hardware. (PGS-TR-175, DAO)

Appendix D.

Disposition of Unused Requirements.

In this Appendix, we present disposition of the requirements which were NOT used as requirements for the PGS Toolkit in the body of this document.

In Table D-1:

"Requirement" is the text of the potential requirement, as given by the instrument team.

"Instr/Or" refers to the instrument team or organization that initiated the requirement. In this column, "JPL" refers to all JPL-based instrument teams, including AIRS, ASTER, MISR, SeaWinds and NSCAT.

"Trace #" refers to the internal trace identification, assigned to all explicit potential requirements received from the science software teams.

"Explanation" is the reason that the potential requirement was not included in the body of this document as a requirement for the PGS Toolkit.

The table is sorted by instrument team.

Table D-1 Potential Requirements Not Used, with Explanations (1 of 4)

Requirement	Doc #	Trace #	Instr/Org	Explanation
Provide tools to facilitate the parallel execution of science algorithms.	93-065	PGS-TR-033	AIRS	No tools for this function were identified in the baseline cost. In addition, parallel execution considerations are the domain of the scheduling system, not the PGS Toolkit. ECS scheduling sub-system personnel are aware of the issue.
The PGS Toolkit shall contain tools to allow check points at boundaries between PGEs.	93-065	PGS-TR-034	AIRS	This is a natural part of the system; no tool needed. Tools to do error/status reporting will be part of the Toolkit.
Provide tools to support simultaneous accesses to multiple data sets for the level 0 data for multiple times and instruments.	93-065	PGS-TR-036	AIRS	This is a natural part of the system. No tool needed.
Provide simultaneous access to multiple datasets.	93-067	PGS-TR-077	AIRS	This is a natural part of the system; no tool needed.
Provide access to multiple data sets and concurrent access by multiple processes to any file used in read only mode by all of those processes.	93-064	PGS-TR-023	ASTER	This is a natural part of the system. No tool needed.
Provide a means to provide inputs from the command script (for example via namelist input).	93-064	PGS-TR-024	ASTER	This is a natural part of the system; no tool needed. Tools to do this are TBD.
The PGS Toolkit shall contain tools to attach to standard products the following metadata: science software version number; data and time the file was generated, start and end time of the data; north-south latitude range and east-west longitude range.	93-064	PGS-TR-029	ASTER	No tool needed. Covered by PGSTK-0450. The exact content of metadata is still TBD; these suggestions will be taken into account.
Provide the atmospheric transmission package LOWTRAN.	93-064	PGS-TR-030	ASTER	No tools for this function were identified in the baseline cost. Level 3 requirement PGS-1025 refers only to software available as COTS, and not provided by developers. In addition only one instrument team has requested this function.
Provide an SQL language interface to a DBMS.	93-067	PGS-TR-059	CERES	No tools for this function were identified in the baseline cost. In addition, no specific requirements for a DBMS have been identified for more than one instrument team.

Table D-1 Potential Requirements Not Used, with Explanations (2 of 4)

The PGS Toolkit shall provide access to a standard ellipsoid at 30km altitude.	93-073	PGS-TR-122	CERES	No tools for this function were identified in the baseline cost. In addition, no specific requirements for such a tool have been identified for more than one instrument team.
The PGS Toolkit shall contain IMAKE facilities.	93-067	PGS-TR-066	DPFT III	This is a natural part of the system; no tool needed. IMAKE comes bundled with X-Windows, which is part of the POSIX standard.
Provide a means to decommutate ancillary data.	93-066	PGS-TR-041	JPL	This is a natural part of the system; no tool is needed. Ancillary data is assumed decommutated by EDOS before Toolkit access.
Require memory dump files be in standardized format.	93-066	PGS-TR-054	JPL	This is a natural part of the system; no tool needed. Format of memory dump files are characteristic of whatever debugger the developer uses.
Provide an interactive debugger.	93-066	PGS-TR-055	JPL	No tools for this function were identified in the baseline cost. Note that many workstations come equipped with debugging tools.
The PGS Toolkit shall provide edit capability for existing HDF files.	93-023	PGS-TR-087	JPL	This is a natural part of the system; no tool needed. Covered by PGSTK-0270.
Provide basic linked list manipulator tools.	93-023	PGS-TR-093	JPL	No tools for this function were identified in the baseline cost. In addition, this function was requested by only one instrument team, and hence is not suited for a common toolkit.
Provide tools to return spacecraft location and attitude, sun location, and moon location given UTC in double precision seconds.	93-079	PGS-TR-155	JPL	This is a natural part of the system; no tool needed. Covered by PGSTK-1160.
Provide a higher level function than the Unix 'cron' facility, in order to make it easier to use.	93-056	PGS-TR-003	LIS	No tools for this function were identified in the baseline cost. This functionality may occur naturally as part of the Toolkit design; this is still TBD.
Allow for weekly and monthly status report generation in production scripts.	93-056	PGS-TR-004	LIS	Not a Toolkit issue. Requirement has been passed on to the ECS communications sub-system (CSMS).
Provide a means to read into memory a file that is >99% fill data without excess use of memory.	93-056	PGS-TR-010	LIS	No tools for this function were identified in the baseline cost. In addition, this function was requested by only one instrument team, and hence is not suited for a common toolkit.

Table D-1 Potential Requirements Not Used, with Explanations (3 of 4)

Provide a tool to perform a runtime profiling of science algorithms.	93-067	PGS-TR-063	MISR	No tools for this function were identified in the baseline cost.
Provide tools to allow the output of a PGE to remain in memory for use as input to another PGE.	93-067	PGS-TR-068	MODIS	No tools for this function were identified in the baseline cost. This tool has been requested by only one instrument team.
Provide a rich scripting language, which includes the capability to switch off products.	93-067	PGS-TR-069	MODIS	This is a natural part of the system; no tool needed.
Provide a means to share with other instrument teams utility libraries that are generated within a single instrument team.	93-067	PGS-TR-071	MODIS	No tools for this function were identified in the baseline cost. Level 3 requirement PGS-1025 refers only to software available as COTS, and not provided by developers. In addition only one instrument team has requested this function
Provide access to the SYBASE language.	93-067	PGS-TR-076	MODIS	No tools for this function were identified in the baseline cost. In addition, no specific requirements for a DBMS have been identified for more than one instrument team.
Provide tools to fill data gaps and interpolate in Level 0 data by executing a provided gap-filler algorithm.	93-073	PGS-TR-099	MODIS	No tools for this function were identified in the baseline cost. The Toolkit provides access to Level 0 metadata, including any information about data gaps given by EDOS. However, no manipulation of Level 0 data itself is done by the Toolkit.
Provide tools to facilitate parallel processing of science software.	93-073	PGS-TR-101	MODIS	No tools for this function were identified in the baseline cost. In addition, parallel execution considerations are the domain of the scheduling system, not the PGS Toolkit. ECS scheduling sub-system personnel are aware of the issue.
Provide tools to access a DBMS.	93-073	PGS-TR-102	MODIS	No tools for this function were identified in the baseline cost. In addition, no specific requirements for a DBMS have been identified for more than one instrument team.
Ensure continued processing of the science software in the event of an exception abort condition.	93-073	PGS-TR-111	MODIS	Not a Toolkit issue; handled by the scheduling sub-system.
The PGS Toolkit shall contain a tool that allows multiple processes access to a common log file.	93-073	PGS-TR-142	MODIS	This potential requirement conflicts with PGS-TR-083. Decision about which one to use rests with the communications sub-system (CSMS). Not a toolkit issue.

Table D-1 Potential Requirements Not Used, with Explanations (4 of 4)

All UNIX system calls and C and FORTRAN library calls will be available to science software.	93-077	PGS-TR-149	MODIS	The issue of which calls will be allowed is addressed in Appendix B of the Toolkit Specification (POSIX).
Provide access to code checkers, configuration management, and similar tools utilizing the X-11 windows system.	93-076	PGS-TR-163	MODIS	This issue is addressed in "Science Users Guide and Operations Handbook, Volume 4: Algorithm Development and Integration", Hughes, August 1993.
Allow informational messages to pass between S/W of different instruments.	93-076	PGS-TR-168	MODIS	This is a natural part of the system; no tool needed. This can be accomplished using the Toolkit tools that handle Q/A flags.
The PGS Toolkit shall contain Q/A flags which recognize incomplete coverage in the Level 0 data streams and provide an appropriate indication in the file header.	93-078	PGS-TR-172	MOPITT	No tools for this function were identified in the baseline cost. The Toolkit provides access to Level 0 metadata, including any information about data gaps given by EDOS. However, no manipulation of Level 0 data itself is done by the Toolkit.
The PGS Toolkit shall contain tools to ensure that independently executing processes do not write to the same log and error files.	93-068	PGS-TR-083	NASA	This potential requirement conflicts with PGS-TR-142. Decision about which one to use rests with the communications sub-system (CSMS). Not a toolkit issue.

This page intentionally left blank.

Abbreviations and Acronyms

AIT	Algorithm Integration Team
ADC	Affiliated Data Center
AIRS	Atmospheric Infrared Sounder
AST	Algorithm Support Team
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer
CASE	Computer Aided Software Engineering
CERES	Clouds and Earth Radiant Energy System
COTS	Commercial Off-The-Shelf (hardware or software)
CPU	Central Processing Unit
CSMS	Communications and Systems Management Segment (ECS)
DAAC	Distributed Active Archive Center
DADS	Data Archive and Distribution System (ECS)
DAO	Data Assimilation Office
DBMS	Data Base Management System
DCE	Distributed Computing Environment
DID	Data Item Description
DPFT	Data Processing Focus Team
ECI	Earth Centered Inertial
ECMWF	European Center for Medium Range Weather Forecasting
ECR	Earth Centered Rotating
ECS	EOSDIS Core System
EDOS	EOS Data and Operations System
EOC	Earth Observation Center (Japan); EOS Operations Center (ECS)
EOS	Earth Observing System
EOSDIS	EOS Data and Information System
ESDIS	Earth Science Data and Information System
ESN	EOS Science Network
GCTP	Geo-Coordinate Transformation Package
GSFC	Goddard Space Flight Center
HAIS	Hughes Applied Information Systems

HDF	Hierarchical Data Format
I/O	Input/Output
ICC	Instrument Control Center (ECS)
IMS	Information Management System (ECS)
IWG	Investigator Working Group
JPL	Jet Propulsion Laboratory
LIS	Lightning Imaging System
L0-L4	Level 0 through level 4 (processing)
MISR	Multi-Angle Imaging Spectro-Radiometer
MODIS	Moderate-Resolution Imaging Spectroradiometer
MTPE	Mission to Planet Earth
NASA	National Aeronautics and Space Administration
NCSA	National Center for Supercomputer Applications
netCDF	network Common Data Format
NMC	National Meteorological Center
NSCAT	NASA Scatterometer
PGE	Product Generation Executable
PGS	Product Generation System (ECS)
PGSTK	Product Generation System Toolkit
POSIX	Portable Operating System Interface for Computer Environments
QA	Quality Assurance
SCF	Science Computing Facility
SGI	Silicon Graphics International
SMC	System Management Center (ECS)
SRR	System Requirements Review
TBD	To Be Determined
TDRSS	Tracking and Data Relay Satellite System
TRMM	Tropical Rainfall Measuring Mission (joint US -Japan)
UARS	Upper Atmosphere Research Satellite
USGS	United States Geodetic Survey
UTC	Universal Time Coordinate
XPG	Xopen Portability Group

This page intentionally left blank.

Glossary

ALGORITHMS (Science Software) consist of the executable programs for science product generation, source code of these executable programs, job control scripts, and algorithm documentation. Algorithms are the result of a new or updated science algorithm passing through the integration and test process, involving the scientist and the PGS's algorithm integration and test staff. After formal approval, algorithms are delivered by the PGS to the DADS for storage, and are retrieved as needed to support product generation production. Some frequently used algorithms may be kept on line in the PGS.

ALGORITHM UPDATES are delivered to the PGS's integration and test environment by scientists at an SCF. They represent changes to existing production algorithms, or a new algorithm to produce a new Standard Product. Algorithm updates include the source code for the candidate algorithm, its associated documentation, and a job step control skeleton. The source code will be compiled to form an executable program suite as part of the integration and test process. The job step control skeleton contains instructions that control the sequence of execution of, and the interchange of data between programs from the executable program suite. Test data sets and calibration data should also be included.

ANCILLARY DATA refers to any data, other than Standard Products, that are required as input in the generation of a Standard Product. This may include selected engineering data from the EOS platform, as well as non-EOS ancillary data. All ancillary data is received by the PGS from the DADS.

CALIBRATION is the collection of data required to perform calibration of the instrument science data, instrument engineering data, and the spacecraft or platform engineering data. It includes pre-flight calibration measurements, in-flight calibrator measurements, calibration equation coefficients derived from calibration software routines, and ground truth data that are to be used in the data calibration processing routine.

DATA AVAILABILITY SCHEDULE is a schedule indicating the times at which specific data sets will be available from remote DADS, EDOS, the IPs, the ADCs and ODCs for ingestion by the collocated DADS. The schedules are received directly by the PGS.

DATA PRODUCTS consist of Level 0 data or Level 1 through Level 4 data products obtained by the PGS from the collocated DADS. These represent the primary input to the product generation process.

DATA QUALITY REQUEST is a request issued by the PGS to a scientist at an SCF to perform QA of a particular product before future processing or distribution. A time window is applied to the request in keeping with the production schedule.

DIRECTIVES consist of information received by the PGS from the SMC that acts as a final authoritative directive for action. It may include general policies, official procedures, and resolutions of schedule conflicts that have not been resolved with the IMS.

DYNAMIC DATA SETS are those containing parameters whose values change routinely and predictably; i.e. at set intervals in time.

L0 PRODUCTION DATA is the result of L0 processing by EDOS on the raw science and engineering data from the EOS platform. It is data that is time-ordered, with overlaps removed.

L0-L4 DATA PRODUCTS consist of L0 Data Products from the IPs, the ADCs and ODCs, and L1-L4 Standard Products produced in the PGS.

METADATA is data which describes a Standard Product including standard metadata (i.e., algorithm and calibration number, size of product, date created, etc.), algorithm-derived metadata, QA information from the PI's, summary statistics and an audit trail.

ON TIME QA is a response to a data quality request that is received within the established production time window. It is received from a scientist at an SCF. It consists of data which will be used to complete the QA fields of the metadata. Overdue QA responses are sent directly to the DADS.

PRODUCT COORDINATION is the coordination of the receipt, staging, and storage of data necessary to carry out the PGS processing schedule.

PRODUCT ORDER is either a request for the generation of a specific product with an associated time window, a priority processing request, a reprocessing request, or a standing order for a product to be generated on a regular basis with a rough timeline, or changes to standing orders. Product orders are received by the PGS from the IMS.

PRODUCT STATUS DIALOG consists of information to assist the IMS in tracking the status of a product order. The IMS may send to the PGS a request for the status of a product to which the PGS would respond with the current product status. The PGS will send a schedule conflict notice to the IMS if a product request will cause a schedule conflict and the IMS may respond with an adjustment to the time window. Schedule conflicts not resolved at this level are resolved by the SMC. The PGS will send an overdue alert to the IMS if it is clear that a product order will not be met on time.

QUICK LOOK DATA are data received during one TDRSS contact period which have been processed to Level 0 (to the extent possible for data from a single contact). These are data that have been identified as requiring priority processing on the order of a few hours. They are routed to the PGS from the DADS.

QUICK LOOK PRODUCTS are the result of applying science algorithms to quick-look data. These products are expeditiously sent over to the DADS, which directly routes them to the appropriate ICC and field campaign users.

STANDARD PRODUCTS are data products that are routinely produced and identified as normal project deliverables.

STATIC DATA SETS are those containing parameters whose values may change, but not routinely at set intervals in time.

STATUS is information regarding schedules, hardware and software configuration, exception conditions, or processing performance. This information is exchanged with the DADS, and is provided to the SMC. The SMC may also receive information regarding schedule conflicts that have not been resolved with the IMS.

SCHEDULES represent the current sequence of tasks to be executed along with approximate execution times as generated by the PGS scheduler. Copies of these schedules, which are updated frequently, are made available to the IMS, the SMC, and the DADS.

TEST PRODUCTS are science products generated by new or updated algorithms during the integration and test period. Test products are delivered to scientists at an SCF.

TEMPORARY FILE is a file which may exists for the duration of a single PGE, or may exist for some indeterminate time beyond the termination of the PGE which created it.