Clouds and the Earth's Radiant Energy System (CERES)

Data Management System

Software Requirements Document

ERBE-like Averaging to Monthly TOA Fluxes
(Subsystem 3.0)

Release 1
Version 1

John L. Robbins, Lee-hwa Chang, and Tammy L. Olaisen

Lockheed Engineering & Sciences Company (LESC)
144 Research Drive
Hampton, VA 23666

Atmospheric Sciences Division
NASA Langley Research Center
Hampton, Virginia 23681-0001

January 1995

Preface

The Clouds and the Earth's Radiant Energy System (CERES) Data Management System supports the data processing needs of the CERES Science Team research to increase understanding of the Earth's climate and radiant environment. The CERES Data Management Team works with the CERES Science Team to develop the software necessary to support the science algorithms. This software, being developed to operate at the Langley Distributed Active Archive Center, produces an extensive set of science data products.

The Data Management System consists of 12 subsystems; each subsystem represents a stand-alone executable program. Each subsystem executes when all of its required input data sets are available and produces one or more archival science products.

The documentation for each subsystem describes the software design at various significant milestones and includes items such as Software Requirements Documents, Data Products Catalogs, Software Design Documents, Software Test Plans, and User's Guides.

The authors wish to express their appreciation to Mr. James L. Donaldson for his work on the ERBE-like code conversion effort, especially for the development of the resource estimation code, the modeling of data products using StP, and for his S-4 code conversion effort.

LIST OF TABLES

1.0 Introduction

The Clouds and the Earth's Radiant Energy System (CERES) is a key component of the Earth Observing System (EOS). The CERES instruments are improved models of the Earth Radiation Budget Experiment (ERBE) scanner instruments, which operated from 1984 through 1990 on the National Aeronautics and Space Administration's (NASA) Earth Radiation Budget Satellite (ERBS) and on the National Oceanic and Atmospheric Administration's (NOAA) operational weather satellites NOAA 9 and NOAA 10. The strategy of flying instruments on Sun-synchronous, polar orbiting satellites, such as NOAA 9 and NOAA 10, simultaneously with instruments on satellites that have precessing orbits in lower inclinations, such as ERBS, was successfully developed in ERBE to reduce time sampling errors. CERES will continue that strategy by flying instruments on the polar orbiting EOS platforms simultaneously with an instrument on the Tropical Rainfall Measuring Mission (TRMM) spacecraft, which has an orbital inclination of 35 degrees. In addition, to reduce the uncertainty in data interpretation, and to improve the consistency between the cloud parameters and the radiation fields, CERES will include cloud imager data and other atmospheric parameters. The first CERES instrument is scheduled to be launched on the TRMM spacecraft in 1997. Additional CERES instruments will fly on the EOS-AM platforms, the first of which is scheduled for launch in 1998, and on the EOS-PM platforms, the first of which is scheduled for launch in 2000.

1.1 Purpose and Objective of Document

This is the Software Requirements Document (SRD) for the ERBE-like Averaging to Monthly TOA Fluxes Subsystem (CERES Subsystem 3.0), which is part of the CERES Data Management System.

The purpose of this volume is to document the requirements that were followed in converting the ERBE Monthly Time/Space Averaging (MTSA) Subsystem code, the ERBE S-4 Output Product Subsystem code, and the ERBE S-4G Output Product Subsystem code to the CERES/ERBE-like code for Subsystem 3.0 (see CERES Top Level Data Flow Diagram in Reference 1) of the CERES Data Management System.

Those who are to be involved in the implementation and testing of Subsystem 3.0 are the intended audience of this document. The document is also to serve as the definitive reference for the ERBE-like Averaging to Monthly TOA Fluxes software requirements.

This document describes the CERES/ERBE-like requirements to convert ERBE code from the Control Data Corporation (CDC) Cyber computers operating under the Network Operating System (NOS) and from the CONVEX (C-Series supercomputer) computer operating under UNIX to run on a Sun/UNIX platform and to process CERES data. Supplemental information may be found in the list of references.

This document is organized as outlined below.

Section 1.0 Introduction states the purpose and objective of the document and gives a Subsystem overview and description of the general content of the document.

Section 2.0 Requirements Approach and Tradeoffs explains where the software requirements originated and outlines what these requirements are.

Section 3.0 External Interface Requirements identifies the major input/output (I/O) products associated with the Subsystem.

Section 4.0 Requirements Specifications describes the Subsystem software operational and functional requirements, outlines the Subsystem design goals and constraints, and provides resource utilization estimates.

References

Abbreviations, Acronyms, and Symbols

Appendix A Conversion Issues contains specific problems encountered during the NOS to UNIX conversion process.

1.2 System Overview

There are two ERBE-like subsystems. Subsystem 2.0 is the daily processor. Subsystem 3.0 is the monthly processor. The strategy for the ERBE-like subsystems is to process CERES data through the same processing system as ERBE with minimal changes to adapt to the CERES characteristics. These Subsystems will be operational at launch.

CERES Subsystem 3.0, ERBE-like Averaging to Monthly TOA Fluxes, consists of the modified ERBE MTSA processor and the ERBE S-4 Output Product Subsystem code from the Cyber, which was combined with the S-4G Output Product Subsystem code from the CONVEX.

The ERBE-like MTSA (EMTSA) code consists of a processor to process the monthly scanner data from a single satellite, a preprocessor to combine the monthly scanner data from two or three satellites, and a main processor to process the combined data from the combined multiple satellites.

The combined code that generates the CERES/ERBE-like ES-4 and ES-4G is referred to as the ES4_S4G code. This code also consists of a preprocessor and a main processor. The preprocessor creates a direct access file that stores area weighting factors and polar day-night indicators needed by the ES4_S4G main processor (see References 2 and 3).

EMTSA produces daily, monthly hourly, and monthly averages of shortwave and longwave radiant fluxes for clear-sky and total-sky at the top-of-atmosphere (TOA) and surface. The monthly hourly average corresponds to estimates for one specific local hour which are averaged over the 31 days in the month. The monthly averages include the average of the daily averages, referred to as the monthly (day) average, and the average of the monthly hourly averages, referred to as the monthly (hour) average. The EMTSA calculations are performed at a 2.5-deg resolution for data from a single satellite as well as multiple (two or three) satellites. For each satellite and every month, the ERBE-like Daily Data Base (EDDB) processor (Subsystem 2.0) rearranges the time referenced output data from Inversion (Subsystem 2.0) into 36 regionally ordered latitudinal files. These files are used by the EMTSA processor to produce the single satellite averages files which in turn are used as input to the EMTSA multiple-satellite processing. ES4_S4G uses the output data from EMTSA, for a single satellite or multiple satellites, to produce geographical averages of radiant flux and albedo values.

Several types of averages are calculated based on the structure of the ERBE grid system. The first type of average is on a regional level. Data values from higher spatial resolution regions are nested into lower spatial resolution regions. Figure 1-1 shows which 2.5-deg and 5-deg regions are nested into 10-deg region 1. The next level of averaging occurs across latitudinal bands. Data from all the regions in a 2.5-deg, 5-deg, or 10-deg latitudinal band are accumulated to produce 2.5-deg, 5-deg, and 10-deg zonal averages. The final type of average is on a global level, where all the zonal averages for a given resolution are averaged. Several global averages are produced for each satellite or combination of satellites. The algorithms for the ERBE-like Averaging to Monthly TOA Fluxes Subsystem are discussed in the CERES Algorithm Theoretical Basis Document (ATBD) for Subsystem 3.0 (see Reference 1). The applicable ERBE software is described in References 4, 5, and 6.

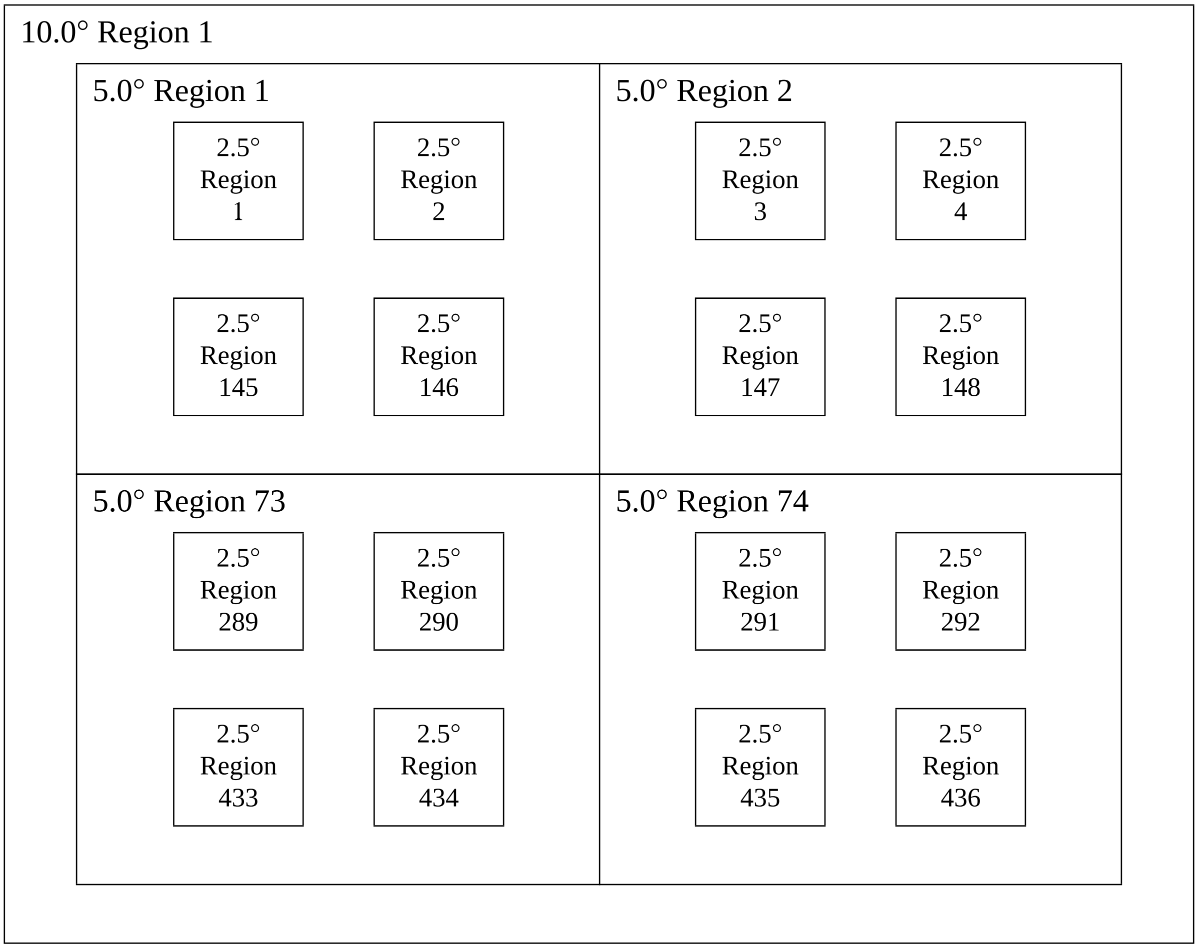

Figure 1-1. Nesting of Regions

Finally, it should be noted that the CERES Science Team has decided not to include surface related parameters on the ERBE-like science products. As a result of that decision, the ERBE-like subsystem names were changed, eliminating the words, "Surface Fluxes." Since this decision was made following both the Formal Inspection and Langley Review of the CERES Subsystem 2.0 (Daily ERBE-like) SRD, that document does not reflect this change.

2.0 Requirements Approach and Tradeoffs

The ERBE-like implementation team formulated the ERBE-like Averaging to Monthly TOA Fluxes software requirements in response to guidance from the CERES Science Team. This guidance was provided through the CERES ATBDs (see Reference 1), meetings of the CERES Science Team, and informal meetings and discussions with members of the CERES Science Team and the Data Management Team.

It should be noted that these software requirements are for converting the original ERBE code to run on a UNIX platform and to process data from the CERES instrument rather than the ERBE instrument, as opposed to a new code development effort. The specific approach is outlined below.

A. Convert the ERBE MTSA Subsystem scanner-only codes to execute on a UNIX platform. Preserve all inputs and outputs as closely to the ERBE design as possible.

B. Convert the ERBE codes used to generate S-4 and S-4G products to a scanner-only configuration that executes on a UNIX platform. Merge the converted code into one processing code. Preserve all inputs and outputs as closely to the ERBE design as possible.

C. Modify the converted scanner-only UNIX code to use Science Data Production (SDP) Toolkit routines as required. Note that this software was formerly referred to as the Product Generation System (PGS) Toolkit.

3.0 External Interface Requirements

This section provides information on the interface requirements which must be satisfied between the ERBE-like Averaging to Monthly TOA Fluxes Subsystem and each of its external input and output products. These interfaces are depicted graphically in the Context Diagram (see Figure 3-1). Each section following the context diagram provides a detailed description of one of the data products represented in the diagram. For a list of every parameter and of sizing estimates for the EDDB, ES-9, ES-4, and ES-4G products, see Reference 1.

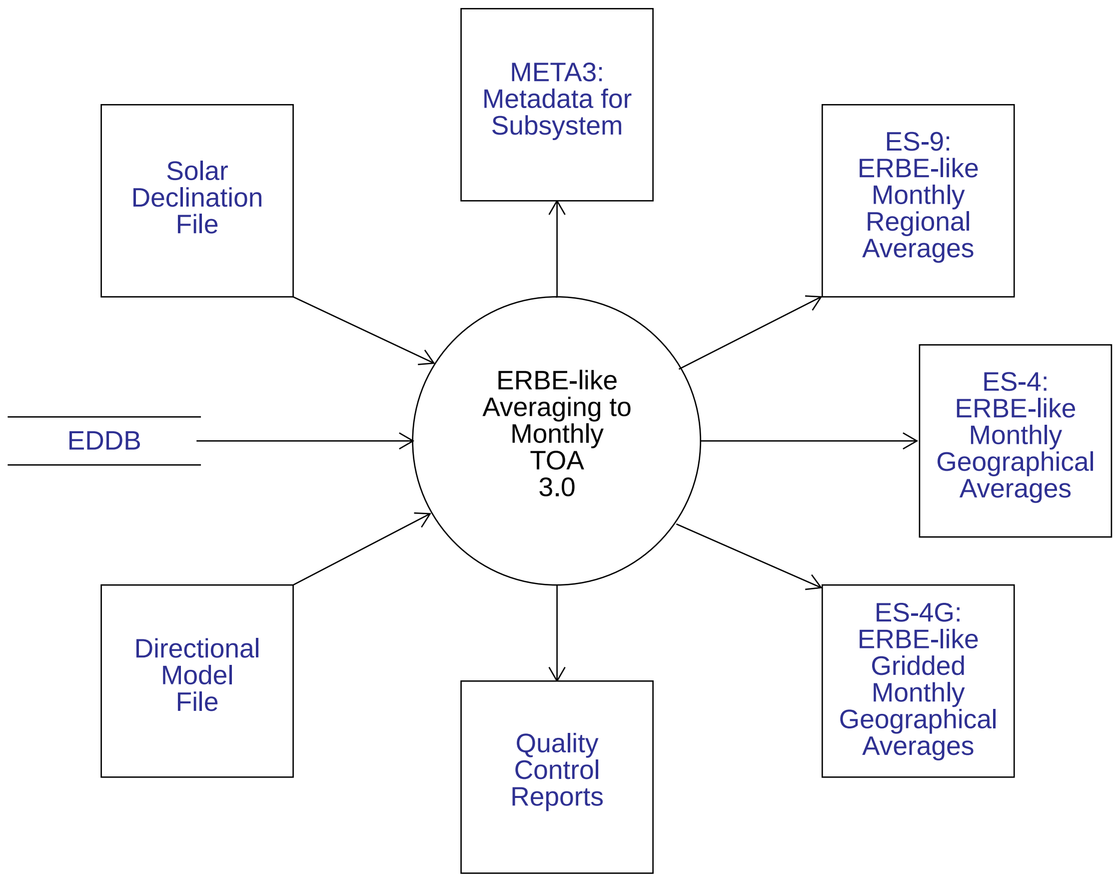

Figure 3-1. Context Diagram

3.1 ERBE-like Daily Data Base (EDDB)

ERBE-like Inversion passes averaged daily regional data to the EDDB processor by way of the ERBE-like EID-6 file. The EDDB processor stores this daily regional data into 36 latitudinal band files for a month. Each of these monthly files contains chronologically arranged data. After accumulating a month of data, the chronologically arranged regional data are sorted by 2.5-deg region number by the EDDB processor. These sorted data are stored in a second set of 36 files for processing by the ERBE-like Monthly Time and Space Averaging Subsystem. Appendix A in Reference 1 describes the parameters contained in the EDDB product.

The EDDB processor also maintains a housekeeping file with information about the status of the regional data. The housekeeping file is a random access file. Each parameter of the housekeeping file corresponds to one or more random access records. The housekeeping and regional data are stored in the EDDB for processing by the ERBE-like Monthly Time and Space Averaging process.

3.2 Directional Model File (ISDM)

This file contains the albedo directional models used by EMTSA in calculating the shortwave TOA estimates. The file is generated whenever the associated information for the ERBE-like Inversion is generated. This procedure insures that the most recent albedo directional model values are used both by daily and monthly ERBE-like subsystems.

3.3 Solar Declination File (TSE)

This file contains the solar constant values and the solar declination angle values for each day of each month for a given year. The file is created off-line and must be generated whenever a new year of data is to be processed or whenever the current file needs to be updated. The same values must be used by all the processors in this Subsystem.

3.4 ERBE-like Monthly Regional Averages (ES-9)

ES-9 stores day-hour box scanner data for all 2.5-deg regions observed during a month for one satellite as well as the inverted daily, monthly hourly, and monthly averages of shortwave and longwave radiant flux at TOA. If more than one satellite is operational during the month, an ES-9 containing the combined multiple satellite data will also be produced.

An ES-9 file contains a scale factors record and the data records. All records in ES-9 are written in 16-bit words. The scale factors are used to pack the data in the data records. There are two data records for each region processed. The first record is of fixed length and contains the averaged data. The second record is of variable length and contains the individual hour box estimates passed from Subsystem 2.0. The ERBE S-9 Users' Guide (see Reference 7) gives the scale factors for each parameter on the ES-9 and shows the format of the data values.

3.5 ERBE-like Monthly Geographical Averages (ES-4)

The major output of the ES4_S4G processor is the ERBE-like Monthly Geographical Averages data file. The ES-4 output data file is structured as follows:

1. Product key and header information. The exact content is to be determined.
2. File address map. The identification numbers, byte addresses, record numbers, and the number of bytes per record in the data are stored in the file address map.
3. Scale factors. The integer scale factors are written in the same order as their corresponding data items. The scale factors are divided into 32-, 16-, and 8-bit words and are used to scale the integer data quantities in each of the data sets.
4. 2.5-deg scanner input data.
5. 2.5-deg nested to 5-deg data.
6. 5-deg nested to 10-deg data.
7. 2.5-deg zonal data.
8. 5-deg zonal data.
9. 10-deg zonal data.
10. Global data.

The scanner input data and the resulting nested, zonal, and global data are grouped into 32-, 16-, and 8-bit words. See Reference 2 for detailed descriptions of the data items.

There are two sets of estimates in an ES-4 file. The first set of estimates uses all scanner measurements, and the second set of estimates uses only those scanner measurements identified as viewing clear-sky areas.

There are three different time periods over which the data are averaged. These are daily, monthly, and monthly hourly. All of these time averages make use of diurnal models (see Reference 2).

3.6 ERBE-like Gridded Monthly Geographical Averages (ES-4G)

The ES-4G output product is an alternative to the ES-4 product. The ES4_S4G processor reorganizes and rewrites the ES-4 data onto four files. Each file contains a variable number of records depending on the data type and resolution. Each record contains a 32-bit word at the beginning and at the end to indicate the total length of that record in bytes. The beginning 32 bits are followed by 80 bytes of ASCII text which describe the parameter, resolution, data type, data date, and spacecraft. The scale factor, data values, and spare values to make each record fall on a 512 boundary follow (see Table 3-1). Therefore, an output record will be structured as shown in Table 3-2.

Table 3-1. ES-4G Files Produced by the ES4_S4G Processor

FILE	NAME[a]	DESCRIPTION	BITS PER WORD	NUM OF RECORDS	DATA VAL PER RECORD	BYTES PER RECORD[b]	BYTES PER FILE
1	S4G1_yymm_s	32-bit Scanner Data[c]	32	87	10368	41984 (420)	3652608
2	S4G2_yymm_s	16-bit Scanner Data[c]	16	342	10368	20992 (166)	7179264
3	S4G3_yymm_s	8-bit Scanner Data[c]	8	221	10368	10752 (295)	2376192
4	S4G4_yymm_s	Nested Averages					5890560
		2.5° nested to 5.0°	32	87	2592	10752 (292)	
			16	342	2592	5632 (358)	
			8	221	2592	3072 (391)	
		5.0° nested to 10.0°	32	87	648	3072 (388)	
			16	342	648	1536 (150)	
			8	221	648	1024 (287)	
		2.5° zonal data	32	87	72	512 (132)	
			16	342	72	512 (278)	
			8	221	72	512 (351)	
		5.0° zonal data	32	87	36	512 (276)	
			16	342	36	512 (350)	
			8	221	36	512 (387)	
		10.0° zonal data	32	87	18	512 (348)	
			16	342	18	512 (386)	
			8	221	18	512 (405)	
		FOV global data	32	87	3	512 (408)	
			16	342	3	512 (416)	
			8	221	3	512 (420)	

a. yymm_s - year month satellite codes.
b. Number in ( ) is number of 8-bit spares for 512 boundary.
c. See Reference 2, Tables 2-1 and 2-2.

Table 3-2. ES-4G Output Record Structure

32-bit word indicating record length ASCII text (80 bytes) scale factor (variable length)
data value 1 (variable length) data value 2 (variable length) • • • data value last (variable length)
spare value 1 (8 bits) spare value 2 (8 bits) • • • spare value last (8 bits)
32-bit word indicating record length

3.7 Metadata for Subsystem 3.0

The contents of this product are to be determined.

3.8 Quality Control Reports

Both EMTSA and ES4_S4G generate rather comprehensive quality control (QC) reports.

EMTSA generates the following QC reports.

1. EQC-9: Combined Longwave and Shortwave Processing Control Vector Summary
2. EQC-34: Statistics Each Channel by Day, Statistics Each Channel by Hour
3. EQC-25: Global Processing Summary for a single satellite
4. EQC-26: Global Processing Summary for combined satellites

The EQC-9 report prints out the processing control vector for up to 20 regions. The EQC-34 report produces the daily, monthly hourly, and grand monthly number of measurements, means, standard deviations, minimums, and maximums for both shortwave and longwave measurements for each of the same regions in the EQC-9 report. Both the EQC-25 and EQC-26 list regions seen by satellite(s) during the observing period.

ES4_S4G generates the following QC reports.

1. EQC-13: 2.5-Degree Regional Summary
2. EQC-14: Zonal Summary
3. EQC-15: Latitudinal Observation Summary
4. EQC-28: ES-4 Output File Summary
5. EQC-50: ES-4G Output File Summary

The EQC-13 reports are generated from the regional pointer array and show the results of the nesting procedure. The EQC-14 reports include a monthly (day) solar incidence value and a monthly (hour) solar incidence value which cover only those periods of time with shortwave measurements in addition to the total solar incidence and the average number of days of shortwave for each latitudinal band. The EQC-15 report lists the average radiant flux values and albedos obtained from the zonal averages. The EQC-28 report gives the number of data files and the number of records in each data file. The EQC-50 report provides similar information for the ES-4G output files.

4.0 Requirements Specifications

4.1 Operating Modes

The CERES Subsystem 3.0 code uses one script for single satellite processing and another script for multiple satellite processing.

The first function performed for each data month by Subsystem 3.0 is the execution of the script for single satellite processing. This is done for each available satellite using the EMTSA single satellite code. This program generates the daily, monthly hourly, monthly (day), and monthly (hour) averages which are passed along to the ES4_S4G program.

The ES4_S4G code uses these averages as input to determine the nested, zonal, and global averages contained on the ES-4 data product. These nested, zonal, and global averages are then used by the ES4_S4G code to generate the gridded ES-4G product.

For multiple satellite processing, the multiple satellite EMTSA code uses ES-9 data, generated by the single satellite code, from two or three satellites as input to produce multiple satellite monthly averages which are passed to the ES4_S4G to generate the ES-4 and ES-4G products in the same way as for single satellite processing. The initial processing and reprocessing, if required, for this Subsystem will be performed in the same way.

4.2 Functional Requirements

The overall strategy for the ERBE-like subsystems, as stated in the Introduction, is to process CERES data through the same processing system as ERBE with only the minimal changes necessary to adapt to CERES characteristics. There are two functional requirements that result from this overall strategy for the ERBE-like Averaging to Monthly TOA Fluxes Subsystem. The first is to port the ERBE MTSA scanner codes for a single satellite and multiple satellites, and the ERBE S-4 and S-4G Output Product codes to a Sun/UNIX platform, to convert the S-4 and S-4G codes to scanner-only, and to merge the scanner-only S-4 and S-4G codes into one processor. The second is to convert the ERBE scanner code to CERES/ERBE-like code. These two requirements are discussed in Sections 4.2.1 and 4.2.2, respectively.

4.2.1 Port ERBE Scanner Code from NOS to UNIX

The first requirement in the development of the CERES/ERBE-like code was to port the necessary ERBE codes to a Sun/UNIX platform. Issues that needed to be dealt with here were the removal of the nonscanner data processing code, 60-bit (NOS) versus 32-bit (UNIX) word sizes and the associated accuracy issues, the porting of several ancillary input data sets needed by the ERBE code, the preparation of replacement code for the ERBE software library routines, and preparation of new script files that would serve the same purpose as the NOS job control language (JCL) procedures.

Appendix A contains more information regarding specific NOS to UNIX conversion issues.

4.2.1.1 Removal of Nonscanner Data Processing Code

ERBE MTSA scanner and nonscanner data processing was performed by two different codes. Consequently, only the scanner codes were ported to UNIX.

For the CERES/ERBE-like ES4_S4G code, the first step taken after the ERBE S-4 and S-4G were ported to a Sun/UNIX platform was to remove the nonscanner data processing code from the ERBE S-4 and S-4G codes. Results from this new UNIX scanner-only code were successfully compared with results from the original NOS codes.

As a result of eliminating the nonscanner processing algorithms, several ERBE output products were also eliminated.

4.2.1.2 60-bit versus 32-bit Word Issues

This issue was addressed by setting all real variables from the original NOS ERBE codes to double precision in the UNIX ERBE-like codes.

4.2.1.3 Ancillary Input Data

The directional model and solar declination ancillary input data files required for EMTSA were ported from the Cyber/NOS to the Sun/UNIX.

4.2.1.4 ERBE Software Library Issues

Tables 4-1 and 4-2 contain the names of ERBE Library (ERBELIB) routines that are used in the ERBE scanner MTSA Subsystem and S-4 and S-4G Output Product Subsystems. The tables also give a brief explanation of how each routine was modified during the conversion process. Source code for these routines was actually incorporated into the ERBE-like codes, so that they could run independently of ERBELIB. The ERBE Data Management System Reference Manual, Volume VIII - System Utilities and User’s Guide (see Reference 6) describes these routines in detail.

Table 4-1. EMTSA Processor ERBELIB Routine Usage

ERBELIB Routine Name	Modifications Due to Conversion
CALJUL	Real data type parameters changed to double precision data types.
DELB	No modifications required.
FINUTL	Changed to only call ABEND and to stop processing in case of a fatal error.
GDAHED	Modified to get product key and IBUF data from EDDB housekeeping file instead of TAPE80.
INUTIL	Modified not to use error message files. Modified to use Sun routines "fdate" and "idate."
JULCAL	Real data type parameters changed to double precision data types.
LENS	No modifications required.
PUTHED	Product key and content of IBUF array written to a specially created holding file instead of TAPE80.
RPTJUL	Real data type parameters changed to double precision data types.
SETLOC	No modifications required. Two additional routines, SETLOC2 and SETLOC8, were created to handle integer*2 data type and double precision data type.
SYSMSG	Changed to print error number, message, and values directly from subroutine where error occurred.
UTLRST	Maximum and minimum values were changed to reflect the new 32-bit word size.
YYMMDD	No modifications required.

Table 4-2. ES4_S4G Processor ERBELIB Routine Usage

ERBELIB Routine Name	Modifications Due to Conversion
CALJUL	Real data type parameters changed to double precision data types.
DELB	No modifications required.
FINUTL	Changed to only call ABEND and to stop processing in case of a fatal error.
INUTIL	Modified not to initialize error message files; these were removed. Modified to use Sun routines "fdate" and "idate".
JULCAL	Real data type parameters changed to double precision data types.
LENS	No modifications required.
REGRPT	Removed use of ERBELIB routine UTLMSG which required the old error message file.
RPTDAT	No modifications required.
RPTJUL	Real data type parameters changed to double precision data types.
SETLOC	No modifications required. An additional routine, SETLOC8, was created to handle double precision data type.
SYSMSG	Changed to print error number, message, and values directly from subroutine where error occurred.
UTLRST	Maximum and minimum values were changed to reflect the new 32-bit word size.
WRHDM	No modifications required.
YYDDD	No modifications required.
YYMMDD	No modifications required.

4.2.1.5 UNIX Scripts

UNIX script files were created to control the execution of the ERBE-like code. The script gets the required input data files according to specified satellite and data date, creates the namelist files of processing and control parameters, invokes the necessary processors, and, finally, saves the output files in the appropriate directories.

4.2.2 Convert ERBE Code to CERES/ERBE-like Code

The second requirement in the development of the CERES/ERBE-like code was to convert the ERBE code to CERES/ERBE-like code. Impacts on the code due to this requirement are primarily modifications necessitated by the required interface with the Distributed Active Archive Center (DAAC) through the SDP Toolkit (see Reference 8). Note that this software was formerly referred to as the PGS Toolkit.

The following is stated in the PGS Toolkit Users Guide for the ECS Project (see Reference 8). "In order to access PGS services such as scheduling and messaging services in a consistent way, to avoid duplication of science software development effort, and to assure portability across computing platforms, usage of a subset of the Toolkit functions is required. These include functions that deal with file I/O, error message transactions, process control, ancillary data access, spacecraft ephemeris and attitude, and time and date transformations. The use of these tools will be enforced through automatic checks at integration time at the DAACs." As indicated in this excerpt from the PGS Toolkit Users Guide for the ECS Project, the impacts to the ERBE-like code will be in the areas of file I/O and error message handling.

I/O files include the primary input to the CERES/ERBE-like monthly code, ancillary input data, processing message files, quality control reports, internal products, and archival data products. The I/O files for the monthly ERBE-like processing are shown in Table 4-3.

Table 4-3. ERBE-like I/O for Subsystem 3.0

Data Type	File	Comment
Internal	EDDB	Primary Input File from the ERBE-like Inversion to Instantaneous TOA and Surface Fluxes Subsystem. Same as ERBE.
Ancillary	ISDM	Directional Model File. Same as ERBE.
Ancillary	TSE	Solar Declination File. Same as ERBE.
Internal	EID-9	Same as ERBE.
Archival	ES-9	Same as ERBE.
Archival	ES-4	Nonscanner parameters have been removed.
Archival	ES-4G	Nonscanner parameters have been removed.

4.2.3 Functional Description of the ERBE Code

The requirements for porting the ERBE code from the Cyber and CONVEX to the Sun and for converting the ERBE code to CERES/ERBE-like code are discussed in Sections 4.2.1 and 4.2.2. The functional requirements for the ERBE code that was the basis for the CERES/ERBE-like code for CERES Subsystem 3.0 are described in detail in References 4 and 5.

4.3 Design Goals and Constraints

Since the ERBE-like Subsystems are converted from the ERBE Subsystems, the primary design goal is to insure that the ERBE-like codes produce the same results as do the ERBE codes with comparable input data. The status of the various steps for achieving that goal is shown in Table 4-4.

Table 4-4. Compliance/Status Matrix for Major Steps in Achieving Design Goals

Goal/Objective	Status
Port the ERBE MTSA, S-4, and S-4G Subsystems codes to Sun/UNIX.	Completed.
Remove non-scanner code from S-4 and S-4G Subsystems codes.	Completed.
Test the scanner-only S-4 and S-4G codes on a Sun/UNIX platform.	Completed.
Merge the scanner-only S-4 and S-4G codes into one program, ES4_S4G.	Completed.
Use SDP Toolkit for Subsystem I/O.	Toolkit interface must be integrated into code and tested.
Be fully operational at launch of the first CERES instrument.	Expect to meet this goal.

It should be remembered that the fundamental guideline from which the ERBE-like software requirements were derived is to develop the capability to process CERES data through the same processing system as was used for ERBE data with only the minimal changes required to adapt to a different computing environment and to the CERES instrument characteristics. This, of course, places limitations on the software system both in terms of recent improvements in software and hardware capabilities beyond that of NOS and in the area of scientific algorithm improvements over ERBE.

4.4 Resource Use

Several parameters have been determined by direct measurement or from system software to evaluate resource usage of the ERBE-like software. This information was obtained by executing the FORTRAN 77 code on a Sun SPARCstation 2 running SunOS 4.1.3 and is summarized in Table 4-5.

Table 4-5. ERBE-like Averaging to Monthly TOA Fluxes Subsystem based on January 1990 ERBS Data

Subsystem	Wall Time (Minutes)	CPU Time (Seconds)	Memory (Kb)
EMTSA (1 Day)	12	676	1288
ES4_S4G (1 Day)	6	283	13436

Sizing information for the primary input and output products can be found in Reference 1.

Sizing information for other I/O products is shown in Table 4-6.

Table 4-6. ERBE-like Subsystem 3.0, Other File Sizing Requirements

File Name	No. of Files Required	Description	Size per File (Bytes)
ISDM	1	Directional model file	1300
TSE	1	Solar declination file	6080
REPTFL	2	Quality Control reports	408499
MSGOUT	2	Error message files	40634
Totals	6		456503

References

1. CERES Algorithm Theoretical Basis Document, ERBE-like Averaging to Monthly TOA and Surface Fluxes (Subsystem 3.0), 1994.

2. ERBE Data Management System Regional, Zonal, and Global Averages S-4 Users' Guide, June 1985.

3. ERBE Data Management System Regional, Zonal, and Global Gridded Averages S-4G User's Guide, November 1990.

4. ERBE Data Management System Reference Manual, Volume VI - DDB and MTSA, 1987.

5. ERBE Data Management System Reference Manual, Volume VII - Output Products, 1986.

6. ERBE Data Management System Reference Manual, Volume VIII - System Utilities and User's Guide, 1986.

7. ERBE Data Management System Earth, Radiant Exitance and Albedo Scanner S-9, Nonscanner S-10 Users' Guides, June 1985.

8. PGS Toolkit Users Guide for the ECS Project (194-809-SD4-001), Version 1 Final, May 1994.

Abbreviations, Acronyms, and Symbols

ATBD	Algorithm Theoretical Basis Document
CDC	Control Data Corporation
CERES	Clouds and the Earth's Radiant Energy System
DAAC	Distributed Active Archive Center
DDB	Daily Data Base
ECS	EOSDIS Core System
EDDB	ERBE-like Daily Data Base (CERES Archival Data Product)
EID-6	ERBE-like Internal Data Product 6 (CERES Internal Data Product)
EID-9	ERBE-like Internal Data Product 9 (CERES Internal Data Product)
EMTSA	ERBE-like Monthly Time/Space Averaging
EOS	Earth Observing System
EOS-AM	Earth Observing System - A.M. platform
EOS-PM	Earth Observing System - P.M. platform
EOSDIS	Earth Observing System Data and Information System
EQC	ERBE-like Quality Control (report)
ERBE	Earth Radiation Budget Experiment
ERBELIB	ERBE Library
ERBS	Earth Radiation Budget Satellite
ES-4	ERBE-like S-4 data product (CERES Archival Data Product)
ES-4G	ERBE-like S-4G data product (CERES Archival Data Product)
ES-9	ERBE-like S-9 data product (CERES Archival Data Product)
ES4_S4G	ERBE-like Regional, Zonal, and Global Averages
FOV	Field of View
I/O	Input and Output
ISDM	Directional Model File
JCL	Job Control Language
MTSA	Monthly Time/Space Averaging
NOAA	National Oceanic and Atmospheric Administration
NOS	Network Operating System
PGS	Product Generation System
QC	Quality Control
SDP	Science Data Production
SRD	Software Requirements Document
TOA	Top-of-Atmosphere
TRMM	Tropical Rainfall Measuring Mission
TSE	Solar Declination File

Appendix A

Conversion Issues

A.1 MTSA Code Conversion Issues

Table A-1 lists most of the major code conversion issues encountered during the NOS to UNIX conversion of the ERBE MTSA Subsystem code.

Table A-1. MTSA Code Conversion Issues

Item No.	Description
1	Real data type variables on NOS were changed to double precision data type.
2	The variables for the 14-digit integer product keys were changed to double precision data type in order to hold the 14-digit values. The array, ibuf, containing the header information from the file was changed to a 9-element array with its first element holding the first 7 digits of the product key and the ninth element holding the remaining 7 digits of the key.
3	In the housekeeping files, the record which held the product key originally was modified to contain its double precision key and the contents of the header buffer, and the records which held the product keys of the daily EID-6 files were modified to hold double precision keys also.
4	The variables of data type integer and real, mixed under one common block on NOS, were rearranged to have the real (currently, double precision on Sun) variables in front of the integer variables since double precision variables use 8-byte boundaries and integer variables use 4-byte boundaries.
5	New routine “setloc8” was used to initialize double precision arrays since the original “setloc” can only be used to initialize integer arrays.
7	Common blocks were stored in separate files and referenced by "include" statements from the program.
8	New routine “resrce” was called to estimate resource usage.

A.2 S-4/S-4G Code Conversion Issues

Table A-2 lists most of the major code conversion issues encountered during the NOS to UNIX conversion of the ERBE S-4/S-4G Subsystem code.

Table A-2. S-4/S-4G Code Conversion Issues

Item No.	Description
1	The nonscanner code of the original S-4 and S-4G was eliminated.
2	The S-4 and S-4G programs were merged into one program.
3	Real data type variables on NOS were changed to double precision data type.
4	The variables for the 14-digit integer product keys were changed to double precision data type in order to hold the 14-digit values. The array, ibuf, containing the header information from the file was changed to a 9-element array with its first element holding the first 7 digits of the product key and the ninth element holding the remaining 7 digits of the key.
5	In the S-4 housekeeping file, the record which held the product key originally was modified to contain its double precision key and the contents of the header buffer.
6	The variables of data type integer and real, mixed under one common block on NOS, were rearranged to have the real (currently, double precision on Sun) variables in front of the integer variables since double precision variables use 8-byte boundaries and integer variables use 4-byte boundaries.
7	The original 7 scanner output files of S-4 were written into one direct access file. A record of this new single file contains a map indicating where in the file the data of the original files were to start. Consequently, the write routine of S-4 and the read routine of S-4G were modified.
8	New routine “setloc8” was used to initialize double precision arrays since the original “setloc” can only be used to initialize integer arrays.
9	Common blocks were stored in separate files and referenced by "include" statements from the program.
10	New routine “resrce” was called to estimate resource usage.