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

Data Management System

Software Requirements Document

ERBE-like Inversion to Instantaneous TOA and Surface Fluxes
(Subsystem 2.0)

Release 1
Version 1

John L. Robbins, Lee-hwa Chang, and James L. Donaldson

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

November 1994

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 one or more stand-alone executable programs. 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 sincere appreciation to Ms. Tammy Olaisen for her invaluable FrameMaker support, particularly in implementing the cross-references and hypertext links and in the book building process, and to the members of the Subsystem 2.0 Formal Inspection Team for finding problems in places where we had forgotten to look. The members of the Subsystem 2.0 Formal Inspection Team included Jim Kibler, Richard Green, Lee-hwa Chang, Sandy Nolan, Lisa Coleman, Keith Degnan, Libby Smith, and Jill Travers.

TABLE OF CONTENTS

TABLE OF CONTENTS (cont'd)

Section Page

Appendices

List of Figures

Figure Page

List of Tables

Table Page

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 for the ERBE-like Inversion to Instantaneous TOA and Surface Fluxes Subsystem, 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 Inversion Subsystem code and the ERBE Daily Data Base (DDB) Subsystem code to the CERES/ERBE-like code for Subsystem 2.0 (see CERES Top Level Data Flow Diagram in Reference 1) of the CERES Data Processing System.

Those who are to be involved in the implementation and testing of Subsystem 2.0 are the intended audience of this document. The document is also to serve as the definitive reference for the ERBE-like Inversion to Instantaneous TOA and Surface Fluxes software requirements.

This document describes the CERES/ERBE-like requirements to convert the ERBE code to run on a UNIX platform and to modify ERBE code 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 and 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 Network Operating System (NOS) to UNIX and ERBE to CERES conversion processes. Also, comments from participants in the ERBE NOS to UNIX conversion experience for the nonscanner-only code are included.

1.2 System Overview

CERES Subsystem 2.0, ERBE-like Inversion to Instantaneous TOA and Surface Fluxes, consists of the ERBE-like Inversion Processor and the ERBE-like Daily Data Base (EDDB) Processor. 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 only minimal changes necessary to adapt to the CERES characteristics. This system will be coded and operational at launch.

The CERES/ERBE-like Inversion code is a version of the original ERBE Inversion Subsystem software which has been modified to process only data from a scanning radiometer. The ERBE Inversion code was further modified to account for the additional scanner samples from the CERES bidirectional scan and to include surface flux calculations. This code converts filtered radiometric measurements in engineering units to instantaneous flux estimates at the top-of-atmosphere (TOA) and at the surface of the Earth. The basis for this procedure is the ERBE processing system which produced TOA fluxes from the ERBE radiometers aboard the ERBS, NOAA-9, and NOAA-10 satellites over a 5-year period from November 1984 to February 1990. The ERBE inversion processing system is a mature set of algorithms that have been well documented and tested. An overview of the ERBE inversion algorithms is given by Smith et al., 1986 (see Reference 2).

The ERBE DDB code was ported from the Control Data Corporation (CDC) Cyber computers operating under NOS to run on a UNIX platform. This code creates a regionally sorted data base for one month of data using the daily time-sorted and inverted data from the ERBE-like Inversion Processor. Other than the modifications required to make it run under UNIX, the EDDB code required little change.

The CERES/ERBE-like algorithms are also discussed in the CERES Algorithm Theoretical Basis Document (ATBD) for Subsystems 2.0 and 3.0 (see References 1 and 3). The applicable ERBE software is described in References 4, 5, and 6.

2.0 Requirements Approach and Tradeoffs

The ERBE-like implementation team formulated the ERBE-like Inversion to Instantaneous TOA and Surface Fluxes software requirements in response to guidance from the CERES Science Team. This guidance was provided through the CERES ATBDs (see References 1 and 3), meetings of the CERES Science Team, and informal meetings and discussions with members of the CERES Science Team and the Data Management Team (DMT).

It should be noted that these software requirements are for converting the original NOS 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 Inversion Subsystem to a scanner-only configuration that executes on a UNIX platform. Modify the resulting code only to cope with source system dependencies. Preserve all inputs and outputs as closely to the ERBE design as possible, including the external products.

B. Convert the ERBE DDB Subsystem to a scanner-only configuration that executes on a UNIX platform. Modify the resulting code only to cope with source system dependencies. Preserve all inputs and outputs as closely to the ERBE design as possible.

C. Modify the converted scanner-only UNIX Inversion FORTRAN code to incorporate the CERES instrument characteristics such as spectral response and increased number of scan samples per input record.

D. Modify the converted scanner-only UNIX Inversion FORTRAN code to input the Meteorological, Ozone, and Aerosol (MOA) ancillary data set and to calculate specified surface parameters.

E. Modify the converted scanner-only UNIX Inversion FORTRAN code to output the new surface parameters to the ES-8 archival data product.

F. Modify the converted scanner-only UNIX Inversion and DDB codes to use Product Generation System (PGS) Toolkit routines as required.

3.0 External Interface Requirements

This section provides information on the interface requirements which must be satisfied between the ERBE-like Inversion to Instantaneous TOA and Surface 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 sizing estimates for the MOA, BiDirectional Scan (BDS), ES-8, and EDDB products, see Reference 1. (It should be noted that in this reference, the MOA is referred to as the Atmospheric Structures (ASTR) Product.)

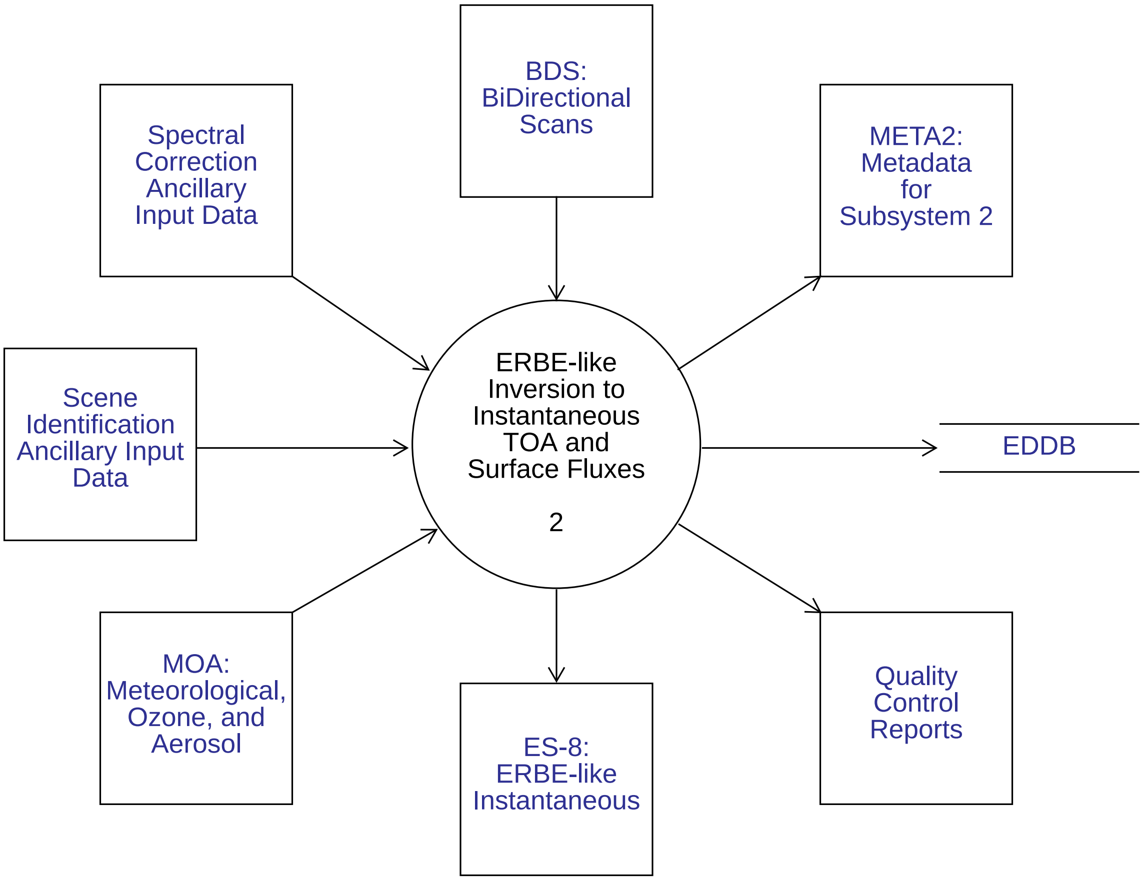

Figure 3-1. Context Diagram

3.1 Spectral Correction Ancillary Input Data

The spectral correction ancillary input data contains parameters and spectral correction coefficients which are used in calculating unfiltered radiances. The spectral correction coefficient data files (see Table 4-8) are instrument dependent. The ERBE-like spectral correction algorithm requires 5760 daytime and 96 nighttime coefficients. For a detailed description of this data set, see Reference 5.

3.2 Scene Identification Ancillary Input Data

The scene identification ancillary input data contains the angular distribution models (ADM) and other ancillary input data required by the ERBE scene identification algorithm as included in the ERBE-like Inversion code. The product is divided into three categories: temporally invariant data, seasonally variant data, and monthly variant data.

The temporally invariant data include the following:

1. Shortwave (SW) bidirectional model, which is a function of scene type and all three viewing angles (viewing zenith, solar zenith, and relative azimuth), and the standard deviation for each SW model value.
2. Normalization constants, which are a function of scene type and solar zenith angle, for linearly interpolated SW bidirectional model values.
3. Mean albedo, which is a function of scene type and solar zenith angle, for each of the SW bidirectional model values.
4. Diurnal change, as a function of scene type, in the longwave (LW) radiant flux as the Sun moves from overhead to midnight.

The seasonally variant data include the following:

1. LW anisotropic model, which is a function of scene type, colatitude, and viewing zenith angle, and the standard deviation for each LW model value.
2. Normalization constants, which are a function of scene type and colatitude, for linearly interpolated LW anisotropic model values.
3. Mean radiant exitance, which is a function of scene type and colatitude, for each of the LW anisotropic models.
4. A priori probability statistics, which are a function of cloud coverage and scene type.
5. Cloud coverage and geo-scene to Inversion scene type mapping values, which are a function of cloud coverage and scene type.

The monthly variant data include the following:

1. Static geographic type, which is a function of colatitude and longitude.
2. Nominal, clear-sky, overhead Sun albedo, which is a function of colatitude and longitude.
3. Nominal, clear-sky, LW radiant exitance, which is a function of colatitude and longitude.

For a detailed description of this data set, see Reference 5.

3.3 Meteorological, Ozone, and Aerosol (MOA)

The CERES archival product, MOA, is produced by the CERES Regrid Humidity and Temperature Subsystem. Each MOA file contains meteorological, ozone, and aerosol data for one hour, and is used by several of the CERES Subsystems. Data on the MOA are derived from several data sources external to the CERES system, such as the National Meteorological Center (NMC), Moderate Resolution Imaging Spectrometer (MODIS), Stratospheric Aerosol and Gas Experiment (SAGE), and various other satellites. These data arrive anywhere from four times daily to once a month and may have entirely different spatial and temporal resolutions than what the CERES system requires. The Regrid Humidity and Temperature Subsystem interpolates these data temporally, horizontally, and vertically to conform with CERES processing requirements.

The MOA contains:

- Surface temperature and pressure
- Vertical profiles for up to 38 internal atmospheric levels of temperature, humidity, and geopotential height as a function of pressure level. The 38 internal atmospheric levels, in hPa, as requested by the CERES Clouds and SARB working groups are listed inTable 3-1
- Column precipitable water
- Vertical ozone profiles at 26 (of the 38) internal atmospheric levels
- Column ozone
- Total column aerosol
- Stratospheric aerosol

Table 3-1. MOA Internal Atmospheric Levels (in hPa)

Floating Levels	**1000 to 875**	**850 to 725**	**700 to 450**	**400 to 225**	**200 to 70**	**50 to 1**
Surface	1000	850	700	400	200	50
Surface - 10	975	825	650	350	175	30
Surface - 20	950	800	600	300	150	10
	925	775	550	275	125	5
	900	750	500	250	100	1
	875	725	450	225	70	

3.4 BiDirectional Scan (BDS)

The BDS data product is an archival product containing level 1b CERES scanner data obtained for a 24-hour period. All science scan modes are included in the BDS, including the fixed and rotating azimuth scan modes that perform normal Earth, internal calibration, and short scan elevation profiles. The BDS product includes samples taken at all scan elevation positions (including space looks and internal calibration views).

The BDS includes the raw count data stream and the converted engineering representative data. These data are divided into the following seven groups that are carried forward from the Level-0 product:

1. Time
2. Instrument Status
3. Radiometric Channel Counts
4. Instrument Telescope Pointing (elevation and azimuth)
5. Temperatures
6. Voltages and Currents
7. Satellite Ephemeris and Ancillary Data

In addition, we add the following filtered radiance data from the three radiometric channels and their associated field-of-view (FOV) location geometry:

8. Filtered Radiances, including quality flags
9. Earth Location Geometry, including quality flags

In the BDS data product, the filtered radiances and the Earth location geometry are considered a multi-band, single data element footprint. Quality flags are used to indicate the reliability of the radiance and Earth location measurements. This product is also used to diagnose instrument performance conditions.

3.5 ERBE-like S-8 (ES-8)

The ES-8 data product contains a 24-hour, single-satellite, instantaneous view of scanner fluxes at the TOA and the surface. These TOA fluxes are reduced from spacecraft altitude unfiltered radiances using the ERBE scanner inversion algorithms and the ERBE SW and LW ADMs. The ES-8 also includes the SW, Total (TOT), and Window (WN) channel radiometric data; SW, LW, and WN unfiltered radiance values; the ERBE scene identification results on a measurement basis; and surface parameters, including SW and LW fluxes, and precipitable water. These data are organized according to the CERES 3.3-sec scan into 6.6-sec records. These records contain only Earth-viewing measurements, approximately 450 for TRMM and 390 for EOS. As long as there is one valid scanner measurement within a record, the ES-8 record will be generated.

The ES-8 is output by the CERES/ERBE-like Inversion program. The TOA and surface fluxes for each CERES measurement will be archived on the ES-8, as well as flags describing instrument status, the radiometric data, and FOV location.

Specifically, the ES-8 contains the following kinds of information:

1. Scan-Level Data
    a) Julian date and time
    b) Earth-Sun distance
    c) Satellite position and velocity
    d) Sun position
    e) Orbit number

2. Measurement-Level Data
    a) Satellite instrument FOV data
    b) Radiometric data
    c) Satellite and Sun geometry data
    d) Unfiltered radiances
    e) TOA fluxes and surface fluxes
    f) ERBE scene identification
        (1) clear ocean
        (2) clear land
        (3) clear snow
        (4) clear desert
        (5) clear coastal
        (6) partly-cloudy ocean
        (7) partly-cloudy land-desert
        (8) partly-cloudy coastal
        (9) mostly-cloudy ocean
        (10) mostly-cloudy land-desert
        (11) mostly-cloudy coastal
        (12) overcast
    g) Precipitable water

The ES-8 will be produced starting at launch and will be externally archived for use by the global scientific community.

3.6 Metadata for Subsystem 2.0 (META2)

The contents of this product are to be determined.

3.7 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 B 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.

3.8 Quality Control Reports

Both the ERBE-like Inversion code and the EDDB code generate rather comprehensive quality control (QC) reports.

The Inversion code generates the EQC-7 report which depicts daily processing information in the following categories:

1. Miscellaneous daily processing statistics
2. Sampling and scene statistics
3. Scene ID performance
4. TOA estimates
5. SW offsets
6. Product statistics

The EDDB update code generates the EQC-10.1, 11.1, and 12.1 reports, which contain summaries of regions filled, hours filled, number of records processed, and TOA statistics. The overlap code generates the EQC-11 report, which shows the current status of the DDB. The postprocessor code generates the EQC-45 report, which summarizes the results of the final data sort (sort according to region number). Detailed examples of these reports may be found in References 5 and 6.

4.0 Requirements Specifications

4.1 Operating Modes

The CERES/ERBE-like code, like the ERBE scanner code, is designed to process on a measurement basis any BDS input data provided. However, the spatial averaging algorithm (see Reference 1) will not properly handle data from the rotating azimuth scanner (see Section 4.2.2.1). Consequently, code has been incorporated into the Inversion program to prevent rotating azimuth scanner measurements from being processed.

Also, if necessary, the Inversion code may be run using the ES-8 as input.

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. The two functional requirements that result from this overall strategy for the ERBE-like Inversion to Instantaneous TOA and Surface Fluxes Subsystem are to port the ERBE scanner code for the Inversion and DDB Subsystems from NOS to a UNIX platform and to convert this ERBE 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 ERBE code from NOS to a UNIX platform. Issues that needed to be dealt with 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 or regeneration of several ancillary input data sets needed by the ERBE code, 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).

Appendix A contains more information regarding specific NOS to UNIX conversion issues. Also, comments from the ERBE NOS to UNIX conversion experience for the nonscanner-only code are included in Appendix A.

4.2.1.1 Removal of Nonscanner Data Processing Code

The first important step that was taken, while the code was still on NOS, was to remove the nonscanner data processing code from the Inversion Subsystem. Results from this new scanner-only code were compared with results from the original code. After the results of the scanner-only code were validated, they became the baseline for validating the ported UNIX code. In the DDB Subsystem, ERBE scanner and nonscanner data processing was performed by two different sets of codes. Consequently, only the scanner code was ported to UNIX as part of ERBE-like Subsystem 2.0.

As a result of eliminating the nonscanner processing algorithms, several ERBE output products were also eliminated. In addition, the Inversion Postprocessor is not needed for scanner data processing and, thus, is not included in the CERES/ERBE-like code.

4.2.1.2 60-bit versus 32-bit Word Issues

This issue was addressed by setting all real variables to double precision explicitly in the FORTRAN code.

4.2.1.3 Ancillary Input Data

For initial testing, ancillary input data files containing spectral correction coefficients, shortwave bidirectional models, longwave anisotropic models, and geo-scene maps were ported from NOS to UNIX. The FORTRAN code that generates those files has since been ported to a UNIX platform, and the data sets have been regenerated on UNIX systems. Also, the namelist files containing processing and control parameters required to run the ERBE-like code were regenerated on a UNIX system.

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 Inversion Subsystem code and DDB Subsystem code. 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 7) describes these routines in detail. Tables 4-1 and 4-2 also give a brief explanation of how each routine was modified during the conversion process.

Table 4-1. Inversion Subsystem ERBELIB Routine Usage

ERBELIB Routine Name	Modifications Due to Conversion
DELB	No modifications required.
FINUTL	Changed to only call ABEND and stop processing in case of a fatal error.
GBFHED	Routine eliminated. Replaced with in-line FORTRAN read.
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.
RPTDAT	No modifications required.
RPTJUL	Real data type parameters changed to double precision data types.
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.
YYMMDD	No modifications required.

Table 4-2. DDB Subsystem ERBELIB Routine Usage

ERBELIB Routine Name	Modifications Due to Conversion
CALJUL	Real data type parameters changed to double precision data types.
CONJUL	Real data type parameters changed to double precision data types.
DELB	No modifications required.
FINUTL	Changed to only call ABEND and stop processing in case of a fatal error.
GDAHED	Modified to get all required header information directly from the housekeeping file, since TAPE 80 is no longer available.
GETHED	Modified to get all required header information from a specially created holding file.
INUTIL	Modified not to initialize error message files; these have been 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.
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 generated to control the execution of the ERBE-like code. The script gets the primary input data file, the required ancillary input data files according to specified satellite and time designations, and the namelist file of processing and control parameters. The programs are run, and the output files are saved 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 in three areas.

1. Modifications due to the CERES instrument and the instrument platform.
2. Modifications due to new CERES data processing algorithms.
3. Modifications necessitated by the required interface with the Distributed Active Archive Center (DAAC) through the PGS Toolkit (see Reference 8).

4.2.2.1 Modifications Due to the CERES Instrument and Instrument Platform

Table 4-3 shows a comparison of several ***instrument and data processing parameters*** for ERBE and for CERES/ERBE-like.

Table 4-3. Differences between the ERBE and CERES Instrument and Output Product Structures

Item	ERBE	CERES/ ERBE-like
Scan Cycle	4 seconds	6.6 seconds[1]
Earth-viewing Measurements per Scan Cycle	62	450
Record Length	16 seconds (S-8)	6.6 seconds (ES-8)
Measurements per Record	248	450
Records per Day	5400	13091
Elements per Record	3360	8706
Elements per Day	18,144,000	113,970,246
1. CERES employs bidirectional scanning.		

Changes in the code were made to account for these differences.

The CERES scanner is a ***rotating azimuth plane scanner***. Since the ES-8 will not contain data for periods when the scanner is rotating, a status flag must be inserted into the data stream to indicate this condition.

The new ***window channel on the CERES instrument*** and the different ***spectral response of the CERES instrument*** (as opposed to the ERBE instrument) required that changes be made both in the spectral correction algorithm and in the spectral correction coefficients.

The ERBE spectral correction algorithm unfiltered the shortwave, longwave, and total channels into shortwave and longwave unfiltered values. The CERES/ERBE-like spectral correction algorithm will unfilter only the shortwave and total channels into shortwave and longwave unfiltered values. The window channel will be unfiltered as an independent process within the ERBE-like Subsystem. Neither TOA nor surface fluxes will be calculated from the window channel radiances in the ERBE-like Inversion code.

CERES spectral correction coefficients based on estimates of the spectral response of the CERES instrument are being developed at the time of this writing.

4.2.2.2 Modifications Due to New CERES Data Processing Algorithms

The ERBE-like Inversion to Instantaneous TOA and Surface Fluxes Subsystem requires the calculation of shortwave and longwave surface fluxes in addition to fluxes at the TOA. The surface fluxes will be routinely calculated from each TOA estimate as long as all required information is available. The following formulation for the net shortwave flux is from Reference 1.

The net shortwave flux at the surface in W m^{-2} is determined as follows:

$$SW^{net}_{surf} = E_o d^{-2} \mu \left\{ 1 - \frac{C}{\mu} - \frac{D}{\sqrt{\mu}} + \frac{1 - \exp(-\mu)}{\mu} (0.0699 - 0.0683\sqrt{p}) \right.$$
$$\left. - [1 + A + B \ln(\mu) - 0.0273 + 0.0216\sqrt{p}]\, a_{TOA} \right\}$$

where

E_o = solar constant = 1365 W m^{-2}
d = Earth-Sun distance in astronomical units
p = precipitable water in cm.
$\mu = \cos\theta_o$
θ_o = solar zenith angle
a_{TOA} = albedo at TOA = $\dfrac{M_{TOA}}{E_o d^{-2} \mu}$
M_{TOA} = reflected shortwave flux at TOA, W m^{-2}
A = 0.0815
B = 0.0139
C = -0.01124
D = 0.1487

All required input for calculating the net shortwave flux at the surface is available in the Inversion code, except precipitable water. Precipitable water will be made available through the PGS Toolkit from the CERES MOA product.

The remainder of the surface parameters are under development: downward shortwave flux at surface, downward longwave flux at surface, and net longwave flux at surface.

Either calculated values or place holders for the surface fluxes will be included on the ES-8 archival product. Precipitable water values used in the surface flux calculations will also be included on the ES-8 product. Neither will be passed on to Subsystem 3.0.

4.2.2.3 Modifications Necessitated by the Required Interface with the DAAC

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 daily code, ancillary input data, error message files, quality control reports, internal products, and archival data products. These files for daily ERBE-like processing are shown in Table 4-4.

Table 4-4. ERBE-like I/O for Subsystem 2.0

Data type	Product	Comment
Archival	BDS	Primary Input File from the Geolocate and Calibrate Earth Radiances Subsystem.
Ancillary	Spectral Correction Coefficients	Changed from ERBE due to different spectral response of the CERES instrument and the introduction of the longwave window (8 to 12 μ) channel.
Ancillary	SW Bidirectional Models	Same as ERBE.
Ancillary	LW Anisotropic Models	Same as ERBE.
Ancillary	Geo-scene Composite Snow Maps	Same as ERBE.
Internal	Quality Control Reports	Includes regionally averaged estimates of the surface flux parameters. Nonscanner parameters have been removed.
Internal	EID-6	Same as ERBE.
Archival	ES-8	Includes estimates of the surface flux parameters. Nonscanner parameters have been removed.
Internal	EDDB	Same as ERBE.

4.2.3 Functional Description of the ERBE Code

The requirements for the reuse of ERBE code and the implementation into that code of the CERES instrument characteristics in place of the ERBE instrument characteristics are described in Sections 4.2.1 and 4.2.2. The functional requirements of the ERBE code that was the basis for the CERES/ ERBE-like code for CERES Subsystem 2.0 are described in detail in the ERBE Reference Manual (see References 4, 5, and 6).

4.3 Design Goals and Constraints

Since the ERBE-like Inversion to Instantaneous TOA and Surface Fluxes Subsystem has been converted from the ERBE Inversion and DDB Subsystems, the design goals are limited. The status of the major design goals is shown in Table 4-5.

Table 4-5. Compliance/Status Matrix for Major Design Goals and Objectives

Goal/Objective	Status
Port the ERBE Inversion Subsystem and DDB Subsystem codes from NOS to UNIX.	Completed.
Modify the Inversion code for changes in scan cycle time, number of measurements per scan cycle, and the record length of the S-8 versus ES-8.	Completed.
Work BDS flag issues including the new rotating scanner flag with the Instrument Subsystem.	Ongoing.
Modify the CERES/ERBE-like Inversion code and ancillary input data files to handle the new CERES longwave window channel and to account for the spectral response of the CERES instrument.	Design/Code implementation approach is described in Section 4.2.2.1. Modifications to the spectral correction coefficients are being investigated by the Science Team.
Introduce Surface Flux Calculations.	Completed for the net shortwave at the surface flux. There are no specifications from the Science Team for the remaining surface flux calculations for ERBE-like processing.
Use PGS Toolkit for Subsystem I/O.	Toolkit interface must be integrated into code and tested.
Flow ERBE simulation data through the ERBE-like Inversion to Instantaneous TOA and Surface Fluxes Subsystem by June 1994.	Completed; validating results.
Be fully operational at launch of the first CERES instrument.	Expect to meet this goal.

Since there is not a stream of CERES instrument data, the conversion to the ERBE-like Inversion to Instantaneous TOA and Surface Fluxes Subsystem can only be tested for data and logic flow errors. ERBE data has been used to simulate the filtered radiances from the BDS data product, so output results can be evaluated for reasonableness.

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 utilization of the ported ERBE and the ERBE-like software. This information is summarized in Tables 4-6 and 4-7.

Table 4-6. ERBE Inversion and Daily Data Base Subsystems based on January 1990 ERBS Data

Subsystem	Wall Time (Minutes)	CPU Time (Seconds)	Memory (Kb)
Inversion	8	447	1360
DDB Update (1 Day)	1	31	932
DDB Sort (1 Day)	2	58	872

Table 4-7. ERBE-like Inversion to Instantaneous TOA and Surface Fluxes Subsystem based on Simulated TRMM Data

Subsystem	Wall Time (Minutes)	CPU Time (Seconds)	Memory (Kb)
Inversion	44	2590	1308
DDB Update (1 Day)	1	32	868
DDB Sort (1 Day)	2	58	804

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 4-8.

Table 4-8. ERBE-like Subsystem 2.0, Other File Sizing Requirements

File Name	No. of Files Required	Description	Size per File (Bytes)
iisw02	1	Temporally constant SW bidirectional model	109,576
iilw_season	4	Seasonal LW anisotropic model	16,968
iigs_month	12	Monthly geo-snow model	31,112
iisc_platform	3	Instrument dependent spectral correction coefficients	116,661
nipsc02	1	Processing and control parameters	1,848
qc reports	4	Quality control reports	82,169
?	5	Error message files	?
Totals	30		358,334

References

1. CERES Algorithm Theoretical Basis Document, ERBE-like Inversion to Instantaneous TOA and Surface Fluxes (Subsystem 2.0), 1994.

2. Smith, G. L., R. N. Green, E. Raschke, L. M. Avis, J. T. Suttles, B. A. Wielicki, and R. Davies, Inversion methods for satellite studies of the Earth's radiation budget: Development of algorithms for the ERBE mission, Rev. Geophys., 24, 407-421, 1986.

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

4. ERBE Data Management System Reference Manual, Volume Va - Inversion, 1987.

5. ERBE Data Management System Reference Manual, Volume Vb - Inversion, 1987.

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

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

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

Abbreviations, Acronyms, and Symbols

ADM	Angular Distribution Model
ATBD	Algorithm Theoretical Basis Document
BDS	BiDirectional Scan (CERES Archival Data Product)
CDC	Control Data Corporation
CERES	Clouds and the Earth’s Radiant Energy System
DAAC	Distributed Active Archive Center
DDB	Daily Data Base
DMT	Data Management Team
ECA	Earth Central Angle
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)
EOPS	ERBE Operational Processing System
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-7	ERBE-like Quality Control 7 (report)
EQC-10.1	ERBE-like Quality Control 10.1 (report)
EQC-11	ERBE-like Quality Control 11 (report)
EQC-11.1	ERBE-like Quality Control 11.1 (report)
EQC-12.1	ERBE-like Quality Control 12.1 (report)
EQC-45	ERBE-like Quality Control 45 (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-8	ERBE-like S-8 data product (CERES Archival Data Product)
ES-9	ERBE-like S-9 data product (CERES Archival Data Product)
FOV	Field of View
hPa	hectoPascal
I/O	Input and Output
JCL	Job Control Language
LW	Longwave
MOA	Meteorological, Ozone, and Aerosol (ancillary data set)
MODIS	Moderate Resolution Imaging Spectrometer
NMC	National Meteorological Center
NOAA	National Oceanic and Atmospheric Administration
NOS	Network Operating System
PGS	Product Generation System
QC	Quality Control
SAGE	Stratosphere Aerosol and Gas Experiment

SCF	Science Computing Facility
SW	Shortwave
TOA	Top-of-Atmosphere
TOT	Total
TRMM	Tropical Rainfall Measuring Mission
WN	Window

Appendix A

Conversion Issues

A.1 Inversion Code Conversion Issues

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

Table A-1. Inversion Code Conversion Issues

Item No.	Description
1	Real data type variables on NOS were changed to double precision data type on UNIX except for specific elements of the XPAT array. XPAT(2) and XPAT(3) on UNIX contain 32-bit integers for the fraction time and the Earth-Sun distance.
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 buffers containing the header information of the files were changed to 9-element arrays with their first elements holding the first 7 digits of the product keys and the ninth elements holding the remaining 7 digits of the keys.
3	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 UNIX) variables in front of the integer variables since double precision variables use 8-byte boundaries and integer variables use 4-byte boundaries.
4	Code was modified where NOS system calls were used. Some of the NOS system calls, such as TIME, SECOND, GETPARM, etc., were replaced by similar UNIX system calls and some, such as BUFFER IN, BUFFER OUT, UNITCK, etc. were eliminated.
5	Code was modified where system call ICHAR was used. NOS and UNIX have different character set values. There is no system call on UNIX which is similar to NOS system call WTSET. So, a few values in array DYNID were changed instead of a big program change.
6	Input ID-3 and output ID-20 were changed to direct access files.
7	Common blocks were stored in separate files and referenced by "include" statements from the program.
8	A subroutine was written and incorporated into the Inversion code to estimate resource usage.

A.2 Daily Data Base Code Conversion Issues

Table A-2 lists most of the major code conversion issues encountered during the NOS to UNIX conversion of the ERBE Daily Data Base Subsystem code.

Table A-2. Daily Data Base 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	Common blocks were stored in separate files and referenced by "include" statements from the program.
5	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 UNIX) variables in front of the integer variables since double precision variables use 8-byte boundaries and integer variables use 4-byte boundaries.
6	Code was modified where NOS system calls were used. For instance, NOS system routines, such as PF, SM5SORT, SM5ENR, SM5FROM, SM5TO, SM5KEY, SM5RETA, SM5END, were removed or replaced. There is no system call on UNIX to do a sort on a binary file. New code was written to sort the latitudinal files.
7	A subroutine was written and incorporated into the DDB code to estimate resource usage.

A.3 The ERBE Nonscanner NOS to UNIX Conversion

A.3.1 Overview

Following the conversion of the ERBE Nonscanner code from the NOS Cyber to the UNIX platform, comments were solicited from the ERBE Data Management Team regarding how the conversion process could be improved. Section A.3 contains an edited summary of those comments, and Table A-3 contains a list of the contributors.

A.3.2 Contributors

Table A-3. Contributors

Lockheed Engineering & Sciences Company	**Science Applications International Corporation**
James L. Donaldson	Frank E. Martino, III
Sandy K. Nolan	Libby Smith
Lyle J. Ziegelmiller	Robert Wilson
	Helen Yue

A.3.3 Comments

The following is the edited summary of the contributors’ responses.

A.3.3.1 Organize Conversion Early On, Avoid Redundant Effort

1. The necessary libraries (ERBELIB) should have been completely converted and tested prior to the conversion of the individual subsystems. Many of the problems encountered during the testing phase of the conversion were due to errors/inconsistencies in the ERBELIB code. At times, it seemed that the subsystems were being used to debug the ERBELIB subroutines. Analysts might work on one particular bug for a long time only to find that the bug was actually in ERBELIB and not the subsystem’s code.

2. The conversion of a single subsystem to act as a “pathfinder” for the conversion of the rest of the subsystems would have been a good idea. The completed conversion of one of the ERBE subsystems to a fully operational subsystem on the Product Generation System (PGS) could have accomplished the following:
   a) Discovered a multitude of the problems that have affected all subsystems.

b) Helped with the debugging of ERBELIB. Instead of ERBELIB problems affecting every subsystem simultaneously, the pathfinder could have greatly reduced the problems with ERBELIB for other subsystems.
c) Provided much needed experience for the creation of the C shell scripts for each subsystem.
d) Might have developed a template script that each subsystem could modify for their specific needs.
e) Developed new ERBE standards for the conversion of the remaining subsystems.
f) Provided a working subsystem to aid the development, testing and implementation of the ERBE Operational Processing System (EOPS).
g) Provided a working subsystem to aid the development, testing and implementation of the Configuration Management system.
h) Provided a working subsystem to aid the effort to integrate ERBE processing on the PGS.

3. The documentation for Inversion needs an index with page numbers badly!

4. Never begin converting code from one machine with anything other than code from that machine; never pick up where someone else left off. If the halfway converted code has errors, finishing the conversion can take many times longer than starting from scratch.

A.3.3.2 Define Standards and Decide Rules Early

1. Standards for naming conventions and script formats should be created and approved as early as possible, preferably before code that requires them is written.

2. Planning - A general study and discussion of each subsystem by the DMT to suggest expected changes or improvements would have been helpful before any conversion was started. Not enough guidance and direction was given prior to initiating the code conversion. Some things had to be redone as a result. Example 1 - removing all common blocks and using include statements. Example 2 - standardizing after the conversion was done. Had this been done at the beginning, it would not have been necessary to rename all common blocks and change all include statements after the conversion was done.

3. EOPS lacked the necessary requirements and design to properly develop the software. This problem caused serious delays during the development of the software.

A.3.3.3 Keep Portability Issues In Mind When Writing Code

File names were located in the JCL script rather than the FORTRAN code. The JCL script was long, involved, and very overwhelming to a new user. A better approach would be to have the file names defined in the FORTRAN code for the sake of readability. Further, the values for the unit numbers of files should be defined in the code, especially for code which relies so heavily on unit numbers, and does not explicitly declare the file names. Suppose the code contains the following statement:

```
OPEN (IPS7,FORM='UNFORMATTED').
```

It is unclear what value IPS7 contains, and therefore, even less clear what the name of the file being opened is. The JCL script contains a line with the following format:

```
GET,TAPE22=whatever,
```

There is no easy way of knowing that this pertains to the IPS7 (which turns out to have the value of 22). Further, it is not obvious that the IPS7 refers to the ID-3N!! Information should not be so deeply buried, requiring a treasure hunt for each fact to be uncovered.

Unit numbers for FORTRAN reads and writes, as well as constants used more than once in code, should be assigned to a variable either in a namelist, data statement, or near the beginning of the code.

Common block variables should always be ordered by type and the types ordered by size, with the largest size coming first.

A.3.3.4 Machine Dependency Issues

1. Do not write production code that is dependent on a machine-dependent system like Cyber Record Manager. ERBELIB was unnecessarily made to be dependent on direct calls to the Record Manager subsystem, which is inefficient.

2. Do not write code on the target system (for a conversion) that simulates a system dependency from the source machine. For example, writing a BUFFER IN function on UNIX so that a programmer does not have to change his/her code to use READ statements was a mistake.

3. Do not write routines that pack 8-bit and 16-bit data into 32-bit words, when the target system is byte addressable.

4. Many code conversion changes involved I/O. For example, since UNIX writes no end-of-record marks, to keep the same format for output products which were multi-file products (S-4), an individual file was used for each file on a multi-file product. This required the invention of naming schemes and insertion of more file management code (opens, closes, error messages, more passing of parameters to keep track of the names to indicate which file to write to, etc.).

5. Also regarding I/O, the record length was specified in words on one machine and bytes on the other, which required changing many open statements in DDB and S-4.

6. Different machine-word sizes required many changes. For example, all real variables were made double precision. Consequently, the double precision intrinsic functions were used.

7. Determining which variables needed to be changed to double precision was sometimes difficult, as was finding Sun FORTRAN equivalents to NOS Cyber FORTRAN extensions.

8. Differences in character sets and intrinsic functions added to conversion difficulties, as did the task of creating graphics tools on the Sun equivalent to those on the Cyber.

9. Sun IEEE arithmetic is more accurate than CDC floating point.

   For example, float the number, -6805, and divide it by a scale factor (-6805 / 10000.0) such that the resultant is a decimal fraction (-.6805). Multiply the fractional number by a different scale factor, and convert the result to the nearest integer (NINT(-.6805 * 1000.0)) = -681).

   The NOS Cyber will compute -680 instead of -681. On CDC, the problem occurs when the integer is converted to a decimal fraction. If an integer that has 5 in the 1's place is divided by 10, an "exact" result is obtained. But if it is divided by more than 10, say 10000, the result may not be exactly represented by the floating point hardware. In the inexact case, the number may take on the value -.6804998 or -.6805001. It appears that the former is true more than the latter, but both occur.

   On Sun, the results remain exact, so that the NINT(-680.5) = -681. Interestingly, the Sun documentation alludes to problems like this (see Chapter 10, Section 10.2, page 89 of the Sun FORTRAN User's Guide).

A.3.3.5 Invest In Resources to Aid Porting

It would have been helpful to have a debugger on both the source machine and the target machine.

The ftp utility was unreliable. Distortions occurred when ftp'ing from Sun to NOS Cyber, and it would get hung up on certain characters when ftp'ing from NOS Cyber to Sun.

The ACD printer is a total of four miles (round trip) away. It was the only printer available at the time that could print large amounts of information, a necessary step in debugging without a debugger.

A.3.3.6 Miscellaneous

Never do arithmetic calculations with data constants of different precision and expect the resulting numbers to have the precision of the greater precision constant.

A management decision to have the S-10 as a stream of bytes caused many changes to the Monthly Time and Space Averaging Subsystem. As the S-10 has variable length and fixed length records, a scheme to keep track of how many bytes to write at a given time had to be devised and inserted into single, preprocessor, and multiple satellite code. Also, a program to read this new product had to be developed.