PL-TR-94-2244(I)
Environmental Research Papers, No. 1182

MODERATE SPECTRAL ATMOSPHERIC RADIANCE AND TRANSMITTANCE CODE (MOSART).

Volume I: Installation Reference Manual

William M. Cornette
Prabhat Acharya
David Robertson
Gail P. Anderson

19960212 173

7 November 1995

DTIC QUALITY INSPECTED 4

Approved for Public Release; Distribution Unlimited

PHILLIPS LABORATORY
Directorate of Geophysics
AIR FORCE MATERIEL COMMAND
HANSCOM AFB, MA 01731-3010

"This technical report has been reviewed and is approved for publication"

William A.M. Blumberg

Dr. WILLIAM A.M. BLUMBERG, Chief
Simulation Branch
Optical Environment Division

Roger Van Tassel

Dr. ROGER A. VAN TASSEL, Director
Optical Environment Division

This report has been reviewed by the ESC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS).

Qualified requesters may obtain additional copies from the Defense Technical Information Center (DTIC). All others should apply to the National Technical Information Service (NTIS).

If your address has changed, if you wish to be removed from the mailing list, or if the addressee is no longer employed by your organization, please notify PL/IM, 29 Randolph Road, Hanscom AFB, MA 01731-3010. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE
7 November 1995

3. REPORT TYPE AND DATES COVERED
Scientific Interim

4. TITLE AND SUBTITLE
Moderate Spectral Atmospheric Radiance and Transmittance Code (MOSART). Volume I: Installation Reference Manual

5. FUNDING NUMBERS
PR 3054
TA GD
WU 01

6. AUTHOR(S)
William M. Cornette*
Gail P. Anderson
Prabhat Acharya**
David Robertson**

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Phillips Laboratory/GPOS
29 Randolph Road
Hanscom AFB, MA 01731-3010

8. PERFORMING ORGANIZATION REPORT NUMBER
PL-TR-94-2244(I)
ERP, No. 1182

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES * Photon Research Associates, 10350 North Torrey Pines Road, Suite 300, La Jolla, CA 92037-1020; ** Spectral Sciences, Inc., 99 South Bedford St, Burlington, MA 01803-5169

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The Moderate Spectral Atmospheric Radiance and Transfer (MOSART) computer program calculates atmospheric transmission and radiation in the ultraviolet region through the microwave spectral regions (0.2 µm to infinity or 0 - 50,000 cm^{-1}. The spectral resolution is variable from a value of 2 cm^{-1} upward in increments of 1 cm^{-1}. It contains features which have been extracted from the MODTRAN code developed by the Geophysics Division (PL/GPOS) of the Air Force Phillips Laboratory and the APART code developed by Photon Research Associates (PRA). Because of the requirement that MOSART be compatible with various codes used in the SSGM (Strategic Scene Generation Model), the overall structure of this version of MOSART closely follows that of APART. However, MOSART contains all the MODTRAN atmospheric features and is easily used for that code's usual point-to-point calculations.

This volume provides the user with information describing the installation of MOSART. The other volumes describe running the code (Vol. II), technical discussion (Vol. III), and the structure of MOSART (Vol. IV). To provide users with on-line assistance, this volume is available in a series of "html" files that can be viewed using the MOSART Input Builder or the MOSAIC Software.

14. SUBJECT TERMS
Atmospheric Propagation
Backgrounds
Radiance
Modelling
Transmittance

15. NUMBER OF PAGES
62

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	SAR

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF TABLES (continued)

1.0 INTRODUCTION

The MOderate Spectral Atmospheric Radiance and Transfer (MOSART) computer program calculates atmospheric transmission and radiation in the ultraviolet through the microwave spectral regions (0.2 μm to infinity or 0 - 50,000 cm^{-1}). It contains features which have been extracted from the MODTRAN code developed by the Geophysics Division (PL/GPOS) of the Air Force Phillips Laboratory and the APART code developed by Photon Research Associates, Inc. (PRA). MODTRAN is widely used in many different atmospheric studies, both within and without the DoD. Since APART was developed to provide atmospheric calculations for infrared (IR) signature studies of both targets and backgrounds, it has many features that are desirable for large simulation models. Because of the requirement that MOSART be compatible with various codes used in the SSGM (Strategic Scene Generation Model), the overall structure of this version of MOSART closely follows that of APART. However, MOSART contains all the MODTRAN atmospheric features and is easily used for that code's usual point-to-point calculations.

This volume of the User's Manual describes the installation of the various model elements used in MOSART. The other volumes in the User's Manual describe running the code (Vol. II), a technical discussion (Vol. III), and the software structure of MOSART (Vol. IV).

MOSART and its utility programs are written in ANSI X3.9-1978 FORTRAN (FORTRAN 77) and are very portable programs. The source code delivered with MOSART includes:

FPTEST:	Tests machine dependent operations
INSTDB:	Installs direct access binary data bases
MOSART:	Is the main MOSART program
ASCBIN:	Converts binary files to ASCII and vice-versa
BBTEMP:	Converts radiance to equivalent blackbody temperatures
CRFILE:	Assists in preparing the MOSART input file
FACET:	Calculates the emitted and reflected energy from a simple surface
MRFLTR:	Degrades the spectral output using a filter function
PLTGEN:	Makes graphs of the results
SCNGEN:	Creates statistical scenes
TERTEM:	Calculates diurnal terrain material temperatures
VISUAL:	Converts visible radiances to luminances and determines color

In addition, a MOSART Input Builder Graphical User Interface (GUI) and an on-line version of the User Reference Manual are provided.

2.0 DELIVERY FORMAT

The MOSART code is typically delivered in one of two modes:

- Unbundled (i.e., each module is contained in its own file)
- Bundled (i.e., a source file will contain many modules)

The major advantages to the unbundled mode are that

(i) A module is located in only one file, thereby minimizing the amount of changes required during installation; and
(ii) The make utility can easily be used to create and maintain the executable codes.

If you received one mode and wish to convert to the other mode, the following Unix command files are included:

combine.fil: creates bundled files in a directory named combine/
split.fil: moves the bundled files to a directory named combine/ and creates an unbundled version.

2.1 <u>Unbundled MOSART Package</u>

For a Unix computer, a command file (i.e., direc.make) and a set of make files (e.g., Make_mosart) have been included with the program delivery. Furthermore, each routine is delivered in its own file. The tape should be unloaded into the directory in which you wish to have all the executables (i.e., the MOSART program and its related utility programs). Before running the direc.make command file, you should make the following changes:

1. Change the mode of the command file (i.e., direc.make) to executable files (i.e., chmod +x direc.make)

2. Change the Makefiles as required for your machine. Macros are used to facilitate any changes.

   Note: The PLTGEN program is loaded with the libraries

   -lncarg -lncarg_gks -lncarg_loc

   for the NCAR plotting package. You should use the library appropriate for your computer (see comments on PLTGEN).

Running the command file direc.make will create a directory structure consistent with the Makefiles (see Table 1) and move the files into the appropriate directories. The directory listing is shown in Table 2.

Table 1. Unbundled MOSART Directory Structure.

src:	MOSART source files
src/BD:	MOSART BLOCK DATA source files
bin:	Library and object files
ascbin_src:	ASCBIN source files
bbtemp_src:	BBTEMP source files
crfile_src:	CRFILE source files
facet_src:	FACET source files
fptest_src:	FPTEST source files
instdb_src:	INSTDB source files
mrfltr_src:	MRFLTR source files
pltgen_src:	PLTGEN source files
visual_src:	VISUAL source files
data:	All binary data base files
data/ascii:	All ASCII data base files (compressed)
data/NRLDAT:	SAG/NRL data base files
make:	All make files
test:	All test input and output files

If your machine is not a Unix computer, simply proceed with the installation.

2.2 Bundled MOSART Package

The bundled MOSART package includes these FORTRAN files: ASCBIN.f, BBTEMP.f, CRFILE.f, FACET.f, FPTEST.f, INSTDB.f, MACHIN.f, MOSAR1.f, MOSAR2.f, MOSAR3.f, MOSBD1.f, MOSBD2.f, MRFLTR.f, PLTGEN.f, and VISUAL.f. (Note: The suffix for a FORTRAN source file may be different on your machine, e.g., .FOR, .f77). The contents of these files are shown in Tables 3 through 20.

Several machine-dependent routines are used by MOSART and its accompanying utilities. These are subroutines CONFIG, CNSTNT, IOERR, PROPMT, and TITLCR, and block data DEVCBD. These may have to be edited to meet specific machine requirements. These routines plus a few others, including RDSCN, are grouped into one file called MACHIN.f. This file should be edited and compiled into an object module, MACHIN.o, which then should be linked with appropriate object modules to create FPTEST, MOSART, and the utility executables. Section 3 of the Manual describes most of the changes that need to be made to the routines in MACHIN.f.

Table 2. Unbundled MOSART Directory Contents.

```
Makefile          direc.make*

ascbin_src:
ascbin.f     ascbin.fil   convab.f     setfil.f     slitfn.f
tablea.f     tableb.f     tableh.f     tablet.f

bbtemp_src:
bbtemp.f     bbtemp.fil   invplk.f

crfile_src:
afterp.f     ccor.f       cnvjtk.f     crbkgd.f     crfile.f     crfile.fil
crfltr.f     crinpt.f     cruaer.f     cruatm.f     crucld.f     deceqt.f
densm.f      densu.f      dncalc.f     dnet.f       dno.f        dread.f
glatf.f      glob6s.f     globe6.f     gtd6.f       gtd6bk.f     gts6.f
inarbd.f     inbkbd.f     inclbd.f     interp.f     latphi.f     mdri.f
menu.f       msag.f       nrlbd.f      ohcalc.f     old/         pozone.f
rdmdtn.f     sintrp.f     solzen.f     spline.f     splini.f     splint.f
subsol.f     sun.f        tdep.f       tselec.f     vp.f         vtst.f

data:
UFTAPE       global.dat   scenes.dat

facet_src:
facet.f      facet.fil    rough.f      surfac.f

fptest_src:
ckstat.f     flcoll.f     fptest.f     fptest.fil   lrechk.f

instdb_src:
instdb.f     instdb.fil

make:
Make_ascbin   Make_bbtemp   Make_crfile   Make_fptest   Make_instdb
Make_mosart   Make_mrfltr   Make_pltgen   Make_scngen   Make_tertem
Make_visual

mrfltr_src:
atmint.f     atmout.f     bckint.f     gethdr.f     mrfltr.f     mrfltr.fil

pltgen_src:
agutol.f     append.f     pltbd.f      pltdrv.f     pltgen.f     pltgen.fil
rdmsrt.f

src:
BD/          abccl4.f     abhno4.f     abn2o5.f     abscfc.f     absclo.f
absh2o.f     absmol.f     absn2.f      absn2o.f     absno2.f     abso2.f
abso3.f      absso2.f     aecalc.f     aersol.f     ah2o2.f      airtmp.f
ammnia.f     amolsc.f     beauft.f     beta.f       betau.f      binfil.f
bmod.f       bndmlg.f     bndpar.f     bntpth.f     brbndr.f     calcul.f
calend.f     change.f     chkrst.f     chkver.f     chtime.f     cirex.f
cirrus.f     cities.f     cldalt.f     cldlyr.f     cnstnt.f     coat.f
comfnc.f     config.f     couple.f     csphfn.f     dbands.f     dbinit.f
ddif.f       defalt.f     defbck.f     demsxx.f     denair       denwtr.f
depol.f      derf.f       desaer.f     dflt2.f      dflt8.f      direfl.f
direms.f     disend.f     disprn.f     dndr.f       dpldt.f      drtlay.f
dvincr.f     eclgal.f     ehbs10.f     emissv.f     emtref.f     endpt.f
epheml.f     ephems.f     ephtim.f     eqabs.f      equabs.f     equecl.f
esfit.f      evapor.f     even.f       exgals.f     exoatm.f     exotem.f
file1.fil    file2.fil    file3.fil    filopn.f     filrt.f      filter.f
flstat.f     fluxlw.f     fresnl.f     galrad.f     gam.f        gammln.f
```

Table 2. Unbundled MOSART Directory Contents (continued).

gblbck.f	geom.f	getasp.f	getatm.f	getbck.f	getcld.f
getexo.f	getglc.f	getpos.f	getslr.f	getvar.f	getvec.f
h2ocnt.f	haze.f	heyms.f	hlowt.f	horequ.f	horizn.f
htblnc.f	hydrom.f	ibkcnv.f	ibnsrc.f	idaero.f	igtint.f
igtvec.f	indexi.f	indexw.f	indxbk.f	indxsc.f	inicpl.f
inigeo.f	initl.f	integ.f	intr2d.f	ioerr.f	israel.f
istaer.f	kdistr.f	laylw.f	lctrim.f	lenstr.f	lwcase.f
lyrint.f	machin.fil	marine.f	mdlatm.f	mie.f	mieinp.f
miephs.f	mlscat.f	modbck.f	month.f	mosart.f	mrndfl.f
nchaer.f	nchatm.f	nchaze.f	nchsea.f	ncycle.f	nxxpau.f
o2cnt.f	opath.f	opnscr.f	parse.f	partit.f	pfr.f
phfunc.f	phmlse.f	phydro.f	planck.f	planet.f	plmsub.f
poly.f	pralt.f	prcalc.f	pretem.f	profac.f	prompt.f
prthdr.f	pthosb.f	pthtau.f	putcld.f	puthdr.f	putslr.f
rab.f	radfld.f	radtrx.f	radtry.f	rainex.f	rainsp.f
raypth.f	rbe.f	rdfltr.f	rdgbl.f	rdline.f	rdscn.f
refest.f	refrac.f	relhum.f	resolv.f	rshine.f	satur.f
scintl.f	scnrio.f	seaice.f	seatmp.f	seawtr.f	setalt.f
setbck.f	setflg.f	setup.f	shadow.f	shngeo.f	skynoi.f
slpos.f	slrcnt.f	slunar.f	smpcal.f	snowex.f	snowsp.f
soil.f	solar.f	solbnd.f	solrad.f	spclyr.f	shhair.f
sphice.f	sphwtr.f	sprod.f	sptrig.f	srat.f	srcflx.f
srcgeo.f	srcirr.f	srflux.f	srtlay.f	starad.f	stgeom.f
strcn2.f	sumfil.f	supk.f	swat.f	tangpt.f	termpr.f
thcair.f	thcice.f	thcsnw.f	thcwtr.f	titlcr.f	tmpcld.f
tranlw.f	trnsmt.f	turbul.f	udif.f	udlay.f	upcase.f
usrbck.f	usrcld.f	usrdef.f	virial.f	visrh.f	vsa.f
xmconv.f	xpndar.f	xterp.f	zlat.f	zodicl.f	zrohdr.f
zroint.f					
src/BD:					
arsabd.f	arslbd.f	arsxbd.f	atmsbd.f	bkgdbd.f	bkstbd.f
brbnbd.f	cfcbd.f	chrcbd.f	cirrbd.f	cldrbd.f	crosbd.f
devcbd.f	dsrtbd.f	ecosbd.f	emisbd.f	exmlbd.f	file5.fil
file6.fil	glcfbd.f	h2obd.f	hazebd.f	icebd.f	inflbd.f
inptbd.f	lagrbd.f	lunpbd.f	marnbd.f	molnbd.f	molpbd.f
no2bd.f	o2cbd.f	o2uvbd.f	o3cwbd.f	o3hhbd.f	ocntbd.f
phfgbd.f	phhybd.f	phmabd.f	phocbd.f	phrubd.f	phstbd.f
phtrbd.f	phurbd.f	rainbd.f	refrbd.f	scenbd.f	sicebd.f
slr1bd.f	slr2bd.f	slr3bd.f	slr4bd.f	slr5bd.f	snowbd.f
so2bd.f	stmlbd.f	uftpbd.f	upprbd.f	virlbd.f	wtrbd.f
zod1bd.f	zod2bd.f				
scngen_src:					
coeff.f	corf.f	fm2d.f	four.f	gamma.f	gaus.f
knu.f	runif.f	scale.f	scngen.f	scngen.fil	tdfft.f
tileit.f	uni.f				
tertem_src:					
nchter.f	rdusrm.f	tertem.f	tertem.fil		
test:					
cam.ubk	limb.in	limb.tst	lowtrn.in	lowtrn.tst	
modtrn.in	modtrn.tst	shine.in	shine.tst	test.dif	
test.in	test.out	test.tst	test2.in	test2.tst	
test3.in	test3.tst	total.in	total.tst	user1.in	
user1.tst	user2.in	user2.tst			
visual_src:					
color.f	human.f	nrmlz.f	sumit.f	visual.f	visual.fil

Table 3. Contents of MOSAR1.f.

PROGRAM MOSART	REAL FUNCTION CIREX
REAL FUNCTION ABCCL4	SUBROUTINE CIRRUS
REAL FUNCTION ABHNO4	SUBROUTINE CITIES
REAL FUNCTION ABN205	SUBROUTINE CLDALT
REAL FUNCTION ABSCFC	SUBROUTINE CLDLYR
REAL FUNCTION ABSCLO	SUBROUTINE COAT
REAL FUNCTION ABSH2O	REAL FUNCTION COMFNC
SUBROUTINE ABSMOL	SUBROUTINE COUPLE
REAL FUNCTION ABSN2	REAL FUNCTION CSPHFN
REAL FUNCTION ABSN2O	REAL FUNCTION DBANDS
REAL FUNCTION ABSNO2	REAL FUNCTION DDIF
SUBROUTINE ABSO2	SUBROUTINE DEFALT
REAL FUNCTION ABSO3	SUBROUTINE DEFBCK
REAL FUNCTION ABSSO2	SUBROUTINE DEMSXX
SUBROUTINE AECALC	REAL FUNCTION DENAIR
SUBROUTINE AERSOL	REAL FUNCTION DENWTR
REAL FUNCTION AH2O2	REAL FUNCTION DEPOL
REAL FUNCTION AIRTMP	DOUBLE PRECISION FUNCTION DERF
REAL FUNCTION AMMNIA	SUBROUTINE DESAER
REAL FUNCTION AMOLSC	SUBROUTINE DFLT2
SUBROUTINE ASPECT	SUBROUTINE DFLT8
SUBROUTINE ATMPRN	COMPLEX FUNCTION DIREFL
DOUBLE PRECISION FUNCTION BAND	SUBROUTINE DIREMS
SUBROUTINE BBARSL	SUBROUTINE DISEND
REAL FUNCTION BBO3	SUBROUTINE DISPRN
SUBROUTINE BCKCHK	REAL FUNCTION DNDR
SUBROUTINE BCKGND	REAL FUNCTION DPLDT
SUBROUTINE BCKPRN	SUBROUTINE DRTLAY
REAL FUNCTION BDRF	REAL FUNCTION DVINCR
SUBROUTINE BEAUFT	SUBROUTINE ECLGAL
REAL FUNCTION BETA	REAL FUNCTION EHBSL0
REAL FUNCTION BETAU	REAL FUNCTION EMISSV
SUBROUTINE BINFIL	COMPLEX FUNCTION EMTREF
SUBROUTINE BMOD	SUBROUTINE ENDPT
SUBROUTINE BNDMLG	SUBROUTINE EPHEML
SUBROUTINE BNDPAR	SUBROUTINE EPHEMS
SUBROUTINE BNTPTH	DOUBLE PRECISION FUNCTION EPHTIM
SUBROUTINE BRBNDR	SUBROUTINE EQABS
SUBROUTINE CALCUL	SUBROUTINE EQUABS
SUBROUTINE CALEND	SUBROUTINE EQUECL
SUBROUTINE CHANGE	SUBROUTINE ESFIT
SUBROUTINE CHKRST	
SUBROUTINE CHKVER	
SUBROUTINE CHTIME	

Table 4. Contents of MOSAR2.f.

REAL FUNCTION EVAPOR	SUBROUTINE INIGEO
LOGICAL FUNCTION EVEN	SUBROUTINE INITL
REAL FUNCTION EXGALS	SUBROUTINE INTEG
SUBROUTINE EXOATM	SUBROUTINE INTR2D
REAL FUNCTION EXOTMP	SUBROUTINE ISRAEL
SUBROUTINE FILOPN	INTEGER FUNCTION ISTAER
REAL FUNCTION FILTER	SUBROUTINE KDISTR
SUBROUTINE FLSTAT	SUBROUTINE LAYLW
SUBROUTINE FLUXLW	SUBROUTINE LCTRIM
REAL FUNCTION FRESNL	INTEGER FUNCTION LENSTR
REAL FUNCTION GALRAD	CHARACTER*(*) FUNCTION LWCASE
REAL FUNCTION GAM	SUBROUTINE LYRINT
REAL FUNCTION GAMMLN	SUBROUTINE MARINE
SUBROUTINE GBLBCK	INTEGER FUNCTION MDLATM
SUBROUTINE GEOM	SUBROUTINE MIE
SUBROUTINE GETASP	SUBROUTINE MIEINP
SUBROUTINE GETATM	SUBROUTINE MIEPHS
SUBROUTINE GETBCK	SUBROUTINE MLSCAT
SUBROUTINE GETCLD	SUBROUTINE MODBCK
SUBROUTINE GETEXO	INTEGER FUNCTION MONTH
SUBROUTINE GETGLC	SUBROUTINE MRNDFL
SUBROUTINE GETPOS	INTEGER FUNCTION NCHAER
SUBROUTINE GETSLR	INTEGER FUNCTION NCHATM
REAL FUNCTION GETVAR	INTEGER FUNCTION NCHAZE
SUBROUTINE GETVEC	INTEGER FUNCTION NCHSEA
SUBROUTINE H2OCNT	INTEGER FUNCTION NCYCLE
REAL FUNCTION HAZE	SUBROUTINE NXXPAU
REAL FUNCTION HEYMS	REAL FUNCTION O2CNT
REAL FUNCTION HLOWT	SUBROUTINE OPATH
SUBROUTINE HOREQU	SUBROUTINE OPNSCR
SUBROUTINE HORIZN	SUBROUTINE PARSE
SUBROUTINE HTBLNC	REAL FUNCTION PARTIT
SUBROUTINE HYDROM	REAL FUNCTION PFR
INTEGER FUNCTION IBKCNV	SUBROUTINE PHFUNC
INTEGER FUNCTION IBNSRC	REAL FUNCTION PHMLSC
INTEGER FUNCTION IDAERO	SUBROUTINE PHYDRO
INTEGER FUNCTION IGTINT	REAL FUNCTION PLANCK
SUBROUTINE IGTVEC	SUBROUTINE PLANET
COMPLEX FUNCTION INDEXI	SUBROUTINE PLMSUB
COMPLEX FUNCTION INDEXW	DOUBLE PRECISION FUNCTION POLY
SUBROUTINE INDXBK	SUBROUTINE PRALT
INTEGER FUNCTION INDXSC	SUBROUTINE PRCALC
SUBROUTINE INICPL	SUBROUTINE PRETEM

Table 5. Contents of MOSAR3.f.

SUBROUTINE PROFAC	SUBROUTINE SPCLYR
SUBROUTINE PRTHDR	REAL FUNCTION SPHAIR
SUBROUTINE PTHOSB	REAL FUNCTION SPHICE
SUBROUTINE PTHTAU	REAL FUNCTION SPHWTR
SUBROUTINE PUTCLD	SUBROUTINE SPROD
SUBROUTINE PUTHDR	SUBROUTINE SPTRIG
SUBROUTINE PUTSLR	SUBROUTINE SRAT
REAL FUNCTION RAB	SUBROUTINE SRCFLX
REAL FUNCTION RADFLD	SUBROUTINE SRCGEO
REAL FUNCTION RADTRX	SUBROUTINE SRCIRR
REAL FUNCTION RADTRY	REAL FUNCTION SRFLUX
REAL FUNCTION RAINEX	SUBROUTINE SRTLAY
SUBROUTINE RAINSP	REAL FUNCTION STARAD
SUBROUTINE RAYPTH	SUBROUTINE STGEOM
REAL FUNCTION RBE	REAL FUNCTION STRCN2
SUBROUTINE RDFLTR	SUBROUTINE SUMFIL
SUBROUTINE RDLINE	REAL FUNCTION SUPK
COMPLEX FUNCTION REFEST	SUBROUTINE SWAT
DOUBLE PRECISION FUNCTION REFRAC	SUBROUTINE TANGPT
REAL FUNCTION RELHUM	SUBROUTINE TERMPR
SUBROUTINE RSHINE	REAL FUNCTION THCAIR
REAL FUNCTION SATUR	REAL FUNCTION THCICE
REAL FUNCTION SCINTL	REAL FUNCTION THCSNW
SUBROUTINE SCNRIO	REAL FUNCTION THCWTR
REAL FUNCTION SEAICE	REAL FUNCTION TMPCLD
REAL FUNCTION SEATMP	SUBROUTINE TRANLW
COMPLEX FUNCTION SEAWTR	SUBROUTINE TRNSMT
SUBROUTINE SETALT	SUBROUTINE TURBUL
SUBROUTINE SETBCK	REAL FUNCTION UDIF
SUBROUTINE SETFLG	SUBROUTINE UDLAY
SUBROUTINE SETUP	CHARACTER*(*) FUNCTION UPCASE
REAL FUNCTION SHADOW	SUBROUTINE USRBCK
SUBROUTINE SHNGEO	SUBROUTINE USRCLD
SUBROUTINE SKYNOI	SUBROUTINE USRDEF
REAL FUNCTION SLPOS	REAL FUNCTION VIRIAL
REAL FUNCTION SLRCNT	REAL FUNCTION VISRH
REAL FUNCTION SLUNAR	SUBROUTINE VSA
SUBROUTINE SMPCAL	REAL FUNCTION XMCONV
REAL FUNCTION SNOWEX	SUBROUTINE XPNDAR
SUBROUTINE SNOWSP	REAL FUNCTION XTERP
SUBROUTINE SOIL	REAL FUNCTION ZLAT
REAL FUNCTION SOLAR	REAL FUNCTION ZODICL
SUBROUTINE SOLBND	SUBROUTINE ZROHDR
SUBROUTINE SOLRAD	SUBROUTINE ZROINT

Table 6. Contents of MACHIN.f.

SUBROUTINE CNSTNT
REAL FUNCTION ADD
REAL FUNCTION SUB
REAL FUNCTION MUL
REAL FUNCTION DIV
DOUBLE PRECISION FUNCTION DADD
DOUBLE PRECISION FUNCTION DSUB
DOUBLE PRECISION FUNCTION DMUL
DOUBLE PRECISION FUNCTION DDIV
SUBROUTINE CONFIG
SUBROUTINE DBINIT
BLOCK DATA DEVCBD
SUBROUTINE FILRT
CHARACTER*72 FUNCTION IOERR
SUBROUTINE PROMPT
SUBROUTINE RDGBL
SUBROUTINE RDSCN
SUBROUTINE TITLCR

Table 7. Contents of MOSBD1.f.

BLOCK DATA ARSABD
BLOCK DATA ARSLBD
BLOCK DATA ARSXBD
BLOCK DATA ATMSBD
BLOCK DATA BKGDBD
BLOCK DATA BKSTBD
BLOCK DATA BRBNBD
BLOCK DATA CFCBD
BLOCK DATA CHRCBD
BLOCK DATA CIRRBD
BLOCK DATA CLDRBD
BLOCK DATA CROSBD
BLOCK DATA DSRTBD
BLOCK DATA ECOSBD
BLOCK DATA EMISBD
BLOCK DATA EXMLBD
BLOCK DATA GLCFBD
BLOCK DATA HAZEBD
BLOCK DATA H2OBD
BLOCK DATA ICEBD
BLOCK DATA INFLBD
BLOCK DATA INPTBD
BLOCK DATA LAGRBD
BLOCK DATA LUNPBD
BLOCK DATA MARNBD
BLOCK DATA MOLPBD
BLOCK DATA NO2BD
BLOCK DATA O2CBD
BLOCK DATA O2UVBD
BLOCK DATA O3CWBD
BLOCK DATA O3HHBD
BLOCK DATA OCNTBD

Table 8. Contents of MOSBD2.f.

BLOCK DATA PHFGBD
BLOCK DATA PHHYBD
BLOCK DATA PHMABD
BLOCK DATA PHOCBD
BLOCK DATA PHRUBD
BLOCK DATA PHSTBD
BLOCK DATA PHTRBD
BLOCK DATA PHURBD
BLOCK DATA RAINBD
BLOCK DATA REFRBD
BLOCK DATA SCENBD
BLOCK DATA SICEBD
BLOCK DATA SLR1BD
BLOCK DATA SLR2BD
BLOCK DATA SLR3BD
BLOCK DATA SLR4BD
BLOCK DATA SLR5BD
BLOCK DATA SNOWBD
BLOCK DATA SO2BD
BLOCK DATA STMLBD
BLOCK DATA UFTPBD
BLOCK DATA UPPRBD
BLOCK DATA WTRBD
BLOCK DATA ZOD1BD
BLOCK DATA ZOD2BD

Table 9. Contents of ASCBIN.f.

PROGRAM ASCBIN
SUBROUTINE CONVAB
REAL FUNCTION FILTER
SUBROUTINE GETHDR
REAL FUNCTION GETVAR
INTEGER FUNCTION IBNSRC
SUBROUTINE LCTRIM
INTEGER FUNCTION LENSTR
CHARACTER*(*) FUNCTION LWCASE
SUBROUTINE PARSE
SUBROUTINE RDLINE
SUBROUTINE SETFIL
SUBROUTINE SLITFN
SUBROUTINE TABLEA
SUBROUTINE TABLEB
SUBROUTINE TABLEH
SUBROUTINE TABLET
CHARACTER*(*) FUNCTION UPCASE
REAL FUNCTION XTERP
BLOCK DATA INFLBD
BLOCK DATA MOLNBD

Table 10. Contents of BBTEMP.f.

PROGRAM BBTEMP
SUBROUTINE CHTIME
REAL FUNCTION FILTER
SUBROUTINE GETHDR
REAL FUNCTION GETVAR
INTEGER FUNCTION IBNSRC
REAL FUNCTION INVPLK
SUBROUTINE LCTRIM
SUBROUTINE LENSTR
CHARACTER*(*) FUNCTION LWCASE
SUBROUTINE PARSE
SUBROUTINE PUTCLD
SUBROUTINE PUTSLR
SUBROUTINE RDFLTR
SUBROUTINE RDLINE
SUBROUTINE SETFLG
SUBROUTINE SLRCNT
SUBROUTINE SUMFIL
CHARACTER*(*) FUNCTION UPCASE
REAL FUNCTION XTERP
BLOCK DATA CHRCBD
BLOCK DATA CLDRBD
BLOCK DATA INFLBD
BLOCK DATA SOLRBD

Table 11. Contents of CRFILE.f.

PROGRAM CRFILE	SUBROUTINE MDRI
REAL FUNCTION AFTERP	SUBROUTINE MENU
SUBROUTINE CALEND	INTEGER FUNCTION MONTH
REAL FUNCTION CCOR	SUBROUTINE MSAG
SUBROUTINE CHTIME	REAL FUNCTION OHCALC
SUBROUTINE CNVJTK	REAL FUNCTION POZONE
SUBROUTINE CRBKGD	SUBROUTINE RDMDTN
SUBROUTINE CRFLTR	REAL FUNCTION SINTRP
SUBROUTINE CRINPT	REAL FUNCTION SOLZEN
SUBROUTINE CRUAER	SUBROUTINE SPLINE
SUBROUTINE CRUATM	SUBROUTINE SPLINI
SUBROUTINE CRUCLD	SUBROUTINE SPLINT
SUBROUTINE DECEQT	SUBROUTINE SUBSOL
SUBROUTINE DENSM	SUBROUTINE SUN
SUBROUTINE DENSU	REAL FUNCTION TDEP
SUBROUTINE DNCALC	SUBROUTINE TSELEC
REAL FUNCTION DNET	REAL FUNCTION VP
REAL FUNCTION DNO	REAL FUNCTION VTST
SUBROUTINE DREAD	CHARACTER*(*) FUNCTION UPCASE
DOUBLE PRECISION FUNCTION EPHTIM	BLOCK DATA CHRCBD
REAL FUNCTION GETVAR	BLOCK DATA INARBD
SUBROUTINE GLATF	BLOCK DATA INBKBD
REAL FUNCTION GLOB6S	BLOCK DATA INCLBD
SUBROUTINE GLOBE6	BLOCK DATA INFLBD
SUBROUTINE GTD6	BLOCK DATA INPTBD
BLOCK DATA GTD6BK	BLOCK DATA NRLBD
SUBROUTINE GTS6	
INTEGER FUNCTION IGTINT	
SUBROUTINE INTERP	
SUBROUTINE LATPHI	
SUBROUTINE LCTRIM	
INTEGER FUNCTION LENSTR	
CHARACTER*(*) FUNCTION LWCASE	

Table 12. Contents of FACET.f.

PROGRAM FACET
REAL FUNCTION BDRF
SUBROUTINE CHTIME
DOUBLE PRECISION FUNCTION DERF
SUBROUTINE DIREMS
COMPLEX FUNCTION DIREFL
REAL FUNCTION EHBSL0
REAL FUNCTION FILTER
SUBROUTINE FRESNL
SUBROUTINE GETHDR
REAL FUNCTION GETVAR
SUBROUTINE GETVEC
INTEGER FUNCTION IBNSRC
SUBROUTINE LCTRIM
INTEGER FUNCTION LENSTR
CHARACTER*(*) FUNCTION LWCASE
SUBROUTINE PARSE
REAL FUNCTION PLANCK
DOUBLE PRECISION FUNCTION POLY
SUBROUTINE PROFAC
SUBROUTINE PUTCLD
SUBROUTINE PUTSLR
SUBROUTINE RDFLTR
SUBROUTINE RDLINE
COMPLEX FUNCTION REFEST
REAL FUNCTION ROUGH
SUBROUTINE SETFLG
REAL FUNCTION SHADOW
REAL FUNCTION SLRCNT
SUBROUTINE SUMFIL
REAL FUNCTION SURFAC
CHARACTER*(*) FUNCTION UPCASE
REAL FUNCTION XTERP
BLOCK DATA CHRCBD
BLOCK DATA CLDRBD
BLOCK DATA INFLBD
BLOCK DATA SOLRBD

Table 13. Contents of FPTEST.f.

PROGRAM FPTEST
SUBROUTINE CKSTAT
LOGICAL FUNCTION FLCOL1
INTEGER FUNCTION LRECHK
REAL FUNCTION ZSTAT

Table 14. Contents of INSTDB.f.

PROGRAM INSTDB
CHARACTER*(*) FUNCTION UPCASE

Table 15. Contents of MRFLTR.f.

PROGRAM MRFLTR	CHARACTER*(*) FUNCTION LWCASE
REAL FUNCTION AIRTMP	INTEGER FUNCTION MDLATM
SUBROUTINE ATMINT	SUBROUTINE MIEINP
SUBROUTINE ATMOUT	SUBROUTINE MODBCK
SUBROUTINE ATMPRN	INTEGER FUNCTION MONTH
SUBROUTINE BCKINT	SUBROUTINE MRNDFL
SUBROUTINE BCKPRN	INTEGER FUNCTION NCHAER
SUBROUTINE BEAUFT	INTEGER FUNCTION NCHATM
SUBROUTINE BINFIL	INTEGER FUNCTION NCHAZE
SUBROUTINE CALEND	INTEGER FUNCTION NCHSEA
SUBROUTINE CHKVER	SUBROUTINE PARSE
SUBROUTINE CHKRST	SUBROUTINE PRALT
SUBROUTINE CHTIME	SUBROUTINE PUTCLD
SUBROUTINE CIRRUS	SUBROUTINE PUTSLR
SUBROUTINE DEFALT	SUBROUTINE RDFLTR
SUBROUTINE DFLT2	SUBROUTINE RDLINE
SUBROUTINE DFLT8	REAL FUNCTION RELHUM
REAL FUNCTION DVINCR	REAL FUNCTION SATUR
SUBROUTINE EXOATM	SUBROUTINE SETBCK
EXO REAL FUNCTION EXOTMP	SUBROUTINE SETFLG
SUBROUTINE FILOPN	REAL FUNCTION SLRCNT
REAL FUNCTION FILTER	SUBROUTINE SPTRIG
SUBROUTINE FLSTAT	SUBROUTINE STGEOM
SUBROUTINE GBLBCK	SUBROUTINE SUMFIL
SUBROUTINE GETASP	CHARACTER*(*) FUNCTION UPCASE
SUBROUTINE GETATM	SUBROUTINE USRBCK
SUBROUTINE GETBCK	SUBROUTINE USRCLD
SUBROUTINE GETCLD	SUBROUTINE USRDEF
EXO SUBROUTINE GETEXO	SUBROUTINE VSA
SUBROUTINE GETHDR	REAL FUNCTION XMCONV
SUBROUTINE GETPOS	REAL FUNCTION XTERP
SUBROUTINE GETSLR	SUBROUTINE ZROHDR
REAL FUNCTION GETVAR	SUBROUTINE ZROINT
SUBROUTINE GETVEC	BLOCK DATA ATMSBD
INTEGER FUNCTION IBNSRC	BLOCK DATA BKGDBD
INTEGER FUNCTION IDAERO	BLOCK DATA CHRCBD
INTEGER FUNCTION IGTINT	BLOCK DATA CLDRBD
SUBROUTINE IGTVEC	BLOCK DATA EXMLBD
SUBROUTINE INDXBK	BLOCK DATA INFLBD
INTEGER FUNCTION INDXSC	BLOCK DATA INPTBD
SUBROUTINE INITL	BLOCK DATA MOLPBD
SUBROUTINE INTR2D	BLOCK DATA SCENBD
SUBROUTINE ISRAEL	BLOCK DATA SOLRBD
INTEGER FUNCTION LENSTR	BLOCK DATA STMLBD
SUBROUTINE LCTRIM	BLOCK DATA UPPRBD

Table 16. Contents of PLTGEN.f.

PROGRAM PLTGEN
SUBROUTINE AGUTOL
CHARACTER*(*) FUNCTION APPEND
SUBROUTINE GETHDR
SUBROUTINE LCTRIM
INTEGER FUNCTION LENSTR
SUBROUTINE PLTDRV
SUBROUTINE RDMSRT
SUBROUTINE SLITFN
CHARACTER*(*) FUNCTION UPCASE
BLOCK DATA CHRCBD
BLOCK DATA PLTBD

Table 17. Contents of SCNGEN.f.

PROGRAM SCNGEN
SUBROUTINE COEFF
REAL FUNCTION CORF
SUBROUTINE FM2D
SUBROUTINE FOUR1
REAL FUNCTION GAMMA
REAL FUNCTION GAUS
REAL FUNCTION GETVAR
INTEGER FUNCTION IGTINT
REAL FUNCTION KNU
SUBROUTINE LCTRIM
SUBROUTINE LENSTR
CHARACTER*(*) FUNCTION LWCASE
SUBROUTINE PARSE
SUBROUTINE RDLINE
SUBROUTINE RUNIF
REAL FUNCTION SCALE
SUBROUTINE TDFFT
REAL FUNCTION UNI
CHARACTER*(*) FUNCTION UPCASE

Table 18. Contents of TERTEM.f.

PROGRAM TERTEM
SUBROUTINE ATCALC
REAL FUNCTION DENAIR
REAL FUNCTION DENWTR
REAL FUNCTION EVAPOR
SUBROUTINE GETHDR
REAL FUNCTION GETVAR
SUBROUTINE GETVEC
SUBROUTINE HTBLNC
INTEGER FUNCTION IBNSRC
INTEGER FUNCTION IGTINT
SUBROUTINE INTR2D
SUBROUTINE LCTRIM
INTEGER FUNCTION LENSTR
CHARACTER*(*) FUNCTION LWCASE
INTEGER FUNCTION NCHTER
SUBROUTINE PARSE
REAL FUNCTION PLANCK
SUBROUTINE PROFAC
SUBROUTINE RDLINE
SUBROUTINE RDUSRM
REAL FUNCTION SATUR
REAL FUNCTION SEATMP
SUBROUTINE SPCLYR
REAL FUNCTION SPHAIR
REAL FUNCTION SPHICE
REAL FUNCTION SPHWTR
REAL FUNCTION SRFLUX
REAL FUNCTION THCAIR
REAL FUNCTION THCICE
REAL FUNCTION THCSNW
REAL FUNCTION THCWTR
CHARACTER*(*) FUNCTION UPCASE
REAL FUNCTION VIRIAL
REAL FUNCTION XTERP
BLOCK DATA ATMSBD
BLOCK DATA BKGDBD
BLOCK DATA CHRCBD
BLOCK DATA EXMLBD
BLOCK DATA MOLPBD
BLOCK DATA OCNTBD
BLOCK DATA VIRLBD

Table 19. Contents of VISUAL.f.

PROGRAM VISUAL
SUBROUTINE CHTIME
SUBROUTINE COLOR
SUBROUTINE GETHDR
SUBROUTINE HUMAN
INTEGER FUNCTION IBNSRC
SUBROUTINE LCTRIM
INTEGER FUNCTION LENSTR
SUBROUTINE NRMLZ
SUBROUTINE PUTCLD
SUBROUTINE PUTSLR
SUBROUTINE SETFLG
REAL FUNCTION SLRCNT
SUBROUTINE SUMFIL
SUBROUTINE SUMIT
REAL FUNCTION XTERP
BLOCK DATA CHRCBD
BLOCK DATA CLDRBD
BLOCK DATA SOLRBD

Table 20. Other Files Delivered with MOSART.

combine.fil	Creates a bundled set of files
split.fil	Unbundles the files
Makefile	Calls all Make_* files
test.in	Test input file (see Appendix A)
test.tst	test output file
wrapper.c	C++ "wrapper" for MOSART
clear.c	Sun C++ code to clear error flags

Since the MOSART source code is about 6 megabytes, it has been divided into five smaller files. All of these should be separately compiled and linked with the MACHIN.OBJ file. The following is a list of object modules required for each executable:

MOSART: MOSAR1.o, MOSAR2.o, MOSAR3.o, MOSBD1.o, MOSBD2.o
ASCBIN: ASCBIN.o, MACHIN.o
BBTEMP: BBTEMP.o, MACHIN.o
CRFILE: CRFILE.o, MACHIN.o

FACET:	FACET.o, MACHIN.o
INSTDB:	INSTDB.o, MACHIN.o
FPTEST:	FPTEST.o, MACHIN.o
MRFLTR:	MRFLTR.o, MACHIN.o
PLTGEN:	PLTGEN.o, MACHIN.o, routines from plot library
SCNGEN:	SCNGEN.o, MACHIN.o
TERTEM:	TERTEM.o, MACHIN.o
VISUAL:	VISUAL.o, MACHIN.o

3.0 SOURCE CODE MODIFICATION

If this is the first time that you are installing the MOSART program on a particular type of machine, it is strongly recommended that you use the Floating Point Test (FPTEST) program to check out certain machine-dependent aspects of the MOSART program. This section will explain how to modify the modules for your machine and use FPTEST to test these changes.

The FPTEST program consists of the following modules:

SUBROUTINE CKSTAT: A module for checking wither a code is running in static or dynamic mode; it also checks the setting of uninitialized variables.

SUBROUTINE CNSTNT: A module for performing basic floating point number tests and to set parameters for use throughout the MOSART program.

SUBROUTINE CONFIG: A module where the user can include system-specific routines for configuring the environment.

BLOCK DATA DEVCBD: A module containing a number of system-specific switches, device numbers, filenames, and record lengths.

LOGICAL FUNCTION FLCOLI: A module to determine if a file written by a FORTRAN WRITE can be read by a FORTRAN READ, or if column 1 is suppressed.

PROGRAM FPTEST: A driver for testing the included subroutines and outputting the appropriate results to the screen.

CHARACTER*72 FUNCTION IOERR: A module to provide the user with processor-dependent text messages whenever an input/output error or end-of-file condition occurs.

INTEGER FUNCTION LRECHK: A module that checks the record length for different length and type of records.

SUBROUTINE PROMPT: A module that outputs a character string to the terminal and suppresses the carriage return.

SUBROUTINE TITLCR: A module that obtains the time and date from system-specific routines and creates a title for all MOSART output files.

REAL FUNCTION ZSTAT: A dummy module to assist CKSTAT.

If you have the bundled version, several SUBROUTINEs, several FUNCTIONs, and the BLOCK DATA are also contained in other programs in the MOSART package. If you have to make any changes to them while installing the FPTEST program, you will also have to make identical changes in the MOSART program (see Section 4.0) and its utility programs. Which modules are used in which programs are shown in Table 21. Discussions of potential modifications are presented for each subroutine below.

Table 21. Location of Modules in Programs.

	CONFIG	CNSTNT	PROMPT	TITLCR	DEVCBD	IOERR
FPTEST	*	*	*	*	*	*
INSTDB	*		*		*	*
MOSART	*	*	*	*	*	*
PLTGEN	*		*			*
ASCBIN	*		*		*	*
CRFILE	*		*		*	*
FACET	*	*	*		*	*
MRFLTR	*	*	*		*	*
BBTEMP	*	*	*			*
VISUAL	*	*	*			*

If you have the unbundled version, the changes only have to be made once, in the file containing the module.

3.1 PROGRAM FPTEST

The main PROGRAM is a driver for calling the appropriate SUBROUTINEs, printing the output in user-friendly format, and testing the direct access file I/O. For most computers, no changes have to be made to FPTEST.

The one known exception is the IBM VM/CMS computer, which requires additional information before OPENing a direct access file. The IBM routine required is commented out by 'CIBM'. If you are installing this program on an IBM computer, you will need to replace this string with blanks, not only in FPTEST, but also in the INSTDB and MOSART codes.

3.2 SUBROUTINE CONFIG

This SUBROUTINE provides a method for system-level setting of the configuration of the computer environment. Currently included in CONFIG are the following:

- Floating point error trapping calls for a Sun, commented out by 'CSUN';
- Setting of underflow and read error flags for an IBM VM/CMS operating system, commented out by 'CIBM';

- Setting of an underflow flag for an IBM PC Lahey F77L compiler, commented out by 'CLAH'; and

- Instructions on floating point error trapping for a Silicon Graphics.

3.2.1 Sun Floating Point Error Trapping

For the Sun, this routine must be pre-processed by a C compiler, so remove the string 'CSUN' so that '#' is in column 1, and then change the file name from '***.f' to '***.F'. This code disables the IEEE floating point error handling for a Sun. Several options are available:

First argument:	action	'get'
		'set'
		'clear'
Second argument:	exception	'inexact'
		'division'
		'underflow'
		'overflow'
		'invalid'
		'all' (all five of above)
		'common' (invalid, overflow, division)
Third argument:	address of a signal-handling routine	
		SIGFPE_ABORT
		SIGFPE_IGNORE
		SIGFPE_DEFAULT

For further details, please see the man pages for "ieee_handler." This first pair of SUBROUTINE calls uses a user-defined routine included (and commented out by 'CSUN' at the end of this SUBROUTINE which stops execution if a floating point error is encountered. This second pair of SUBROUTINE calls uses the system signal-handling routines.

3.2.2 IBM VM/CMS Error Disabling

For the IBM VM/CMS, the code disables the warning and error messages and the corrective measures for the IBM VM/CMS operating system:

Error code 208 is for underflow messages
Error code 215 is for READ errors

3.2.3 Lahey F77L Underflow Disabling

For the Lahey F77L FORTRAN compiler on a PC, an underflow results in NDP errors, unless the appropriate system subroutine is executed.

3.2.4 Silicon Graphics Floating Point Error Trapping

For the Silicon Graphics IRIS Workstation, the program must be compiled with a floating point error library:

```
f77 -g -o mosart mosart*.f /usr/lib/libfpe.a
```

The MOSART program is used as an example here. Any program name can be used. The floating point error traps must then be set before executing the program. This change will trap floating point errors for overflows, invalid operations, and divide by zero. All underflows will be ignored and set to 0.0. Overflow will be set to NaN to avoid multiple overflows.

```
setenv TRAP_FPE "DEBUG; OVERFL =INF,COUNT,TRACE(10),EXIT(100);
                       INVALID=IEEE,COUNT,TRACE(10),EXIT(100);
                       DIVZERO=INF,COUNT,TRACE(10),EXIT(100)"
```

The program is then executed. After execution, turn off trapping of floating point errors:

```
setenv TRAP_FPE="OFF"
```

3.2.5 Other Systems

If you have any such routines, either for another type of computer system, or for your particular installation, it is strongly recommended that you place it in this subroutine. If you feel that such routines are of general enough interest, please contact the author.

3.3 SUBROUTINE CNSTNT

A number of machine dependent parameters are calculated using various algorithms. Any machine will have slightly different methods for handling its floating point representations and arithmetic. The MOSART program was written on a machine that complies with the IEEE Standard 754. To compensate for other machines, MOSART uses the SUBROUTINE CNSTNT to define a number of variables that are used throughout the calculations.

The program to perform these calculations uses ANSI X3.9-1978 code with two (2) exceptions:

1. One variable is declared INTEGER*2 and EQUIVALENCEd with an INTEGER variable to determine whether the floating point representation is little-endian or big-endian.

2. The MIL-STD-1753 function IBITS is used to determine if the representation is a one's, two's, or signed complement. For VAX and Lahey PC systems, the

JIBITS function is included but commented out with CVAX and CLAH, respectively. For IBM, a separate FUNCTION IBITS is included using bit manipulation routines available on the IBM system. This code is commented out by "CIBM."

If either of these features is not available on you machine, you can comment out the appropriate code and simply set the variables for your machine, as indicated.

If your machine has routines for providing some of these parameters (e.g., the smallest REAL and DOUBLE PRECISION positive floating point number), these routine can be used and several are included as examples (e.g., FLMIN(), FLMAX(), DFLMIN(), and DFLMAX() for an SGI; R_MIN_NORMAL, R_MAX_NORMAL, D_MIN_NORMAL, and D_MAX_NORMAL for a Sun). If you wish, you can also hard code (rather than calculate) any of these parameters (e.g., see values commented out by 'CGEN').

For VAX machines, it is strongly recommended that you compile the MOSART program with the G-float option, so that the range spanned by the minimum and maximum positive REAL numbers is increased, as well as the number of significant figures.

At the end of CNSTNT are several REAL and DOUBLE PRECISION FUNCTIONs that are used to insure that the calculations are performed with intermediate steps placed in storage, rather than in registers. This is done since some machines (e.g., the Lahey PC) retain greater precision if all the calculations are performed using registers.

3.4 SUBROUTINE PROMPT

The MOSART program and its utility programs use the SUBROUTINE PROMPT to output a message to the terminal and to suppress the carriage return. In SUBROUTINE PROMPT, there are three (3) different FORMAT statements for accomplishing this carriage return suppression:

1. FMT='(1X/1X,A,$)'
2. FMT='(1X/"$",A)'
3. FMT='(1X/1X,A\)

The program is delivered with the first FORMAT implemented and the other two commented out. If none of these work on your computer, either insert a FORMAT that does or use the non-suppressing FORMAT, namely

FMT='(1X/1X,A)'

which is also commented out.

3.5 SUBROUTINE TITLCR

To assist the user in identifying the time and date of the creation of any MOSART file, the time and date are placed in the variable TITLE, using the SUBROUTINE TITLCR. Since the ability to access the time and date is machine dependent (at least for FORTRAN 77), this SUBROUTINE must be modified for the computer on which it is being installed. The MOSART program, as delivered, is designed to operate on a Unix machine using a call to the CHARACTER FUNCTION FDATE, with the appropriate declarations. Declarations and program for other computers are commented out with descriptive character strings:

CVAX	-	VAX VMS
CUNX	-	Unix SUBROUTINE FDATE (the FUNCTION FDATE is implemented on the delivered program; some Unix machines use a SUBROUTINE rather than a FUNCTION)
CLAH	-	Lahey PC
CIBM	-	IBM VM/CMS
CPRI	-	Primos
CCDC	-	CDC NOS
CRS6	-	RS/6000 machine
CF90	-	FORTRAN 90 standard call

Also, for those machines for which the time and date are not readily available (or known to the programmer), a generic capability is provided (commented out by 'CGEN') which loads blanks into the location for the time and date.

If your computer requires code that is not currently provided in TITLCR, please notify the author so that it can be incorporated into future versions of MOSART.

3.6 BLOCK DATA DEVCBD

This module contains a number of device and installation dependent variables that need to be properly set before compiling and executing the program.

3.6.1 File Suffixes

All the output files and user-supplied input files for the MOSART program and its utility programs are assumed to be in form of a file name (supplied by the user at the time of execution) and a file root.

For most computers, the file suffixes as supplied are acceptable and are in the form

'fileroot.suffix'

You may wish to tailor these suffixes for the conventions on your machine (e.g., the use of capitals for VAX VMS). The only two known exceptions to this file naming convention are the PC and the IBM VM/CMS operating system. The PC restricts the file suffix to 3 characters, and the IBM VM/CMS operating system uses the form

'/ FILENAME FILETYPE DISK'

For the PC, simply rename the suffixes appropriately. For the IBM VM/CMS convention, the suffix should be modified to include the FILETYPE and DISK. Further instructions for modifying the code to include the '/' are discussed in Section 4.0.

3.6.2 Device Numbers

The device numbers for all output and user-defined input files range between 11 and 27. Scratch files are numbers 28, 29, 30, and 31, while data base file numbers are 32 through 61. If your computer reserves any of these numbers for its own use or places other restrictions on unit numbers, please make the appropriate changes in the variables named IF*** in COMMON block /DEVICE/, where '***' is a three-character string. Also, certain machines are configured to permit only a certain number of files be attached to a single job. If you experience any problems with the number of files, please contact your systems administrator to increase the maximum number allowed or contact the author.

3.6.3 Directory Path and Data Base File Names

The molecular data bases, the global scene/altitude data base, the global climatology data base, and the SAG/NRL data bases are installed with a directory path and file names that are built into the program. It is strongly recommended that the directory path be as global as possible to permit the MOSART program to be executed from anywhere in the computer directory system. If it is possible to define these files for multiple simultaneous READs (see comments in Section 4.0 for installation of the MOSART program), this will permit the use of MOSART by multiple users. The variable DIRPTH is used as a common directory path for all the binary data bases (see discussion on UFTAPE below). The file name is appended to DIRPTH to specify the global location of the file. It is recommended that the all the file names be local, while the global part of the address be in DIRPTH. This allows only a single change to the program if the data bases have to be moved to another directory.

The convention used by the MOSART program is to OPEN the file if the name is provided and the file exists. A file name of 'None' will notify the MOSART program that the file is not to be used in any calculations. Also, if disk space is a problem, either one or both of the global data bases can be deleted; however, if this is done, the MOSART program will use default values.

According to the ANSI X3.9-1978 Standard, the file name must be a CHARACTER expression whose value, when any trailing blanks are removed is a file name. The file name must be a name that is allowed by the processor. Please refer to your computer manuals for the allowable format for a file name.

3.6.4 Direct Access Record Length

The record sizes of the direct access binary data files (i.e., molecular data bases, global data base, scene data base) are computer dependent. Some machines use the number of bytes, while others use the number of words. Some machines (e.g., IBM VM/CMS) require that a certain number of bytes be added to the record length of direct access files.

There are three types of direct access data files used by the MOSART program:

Molecular data bases:	48 words or 192 bytes per record (IRECL)
Global scene/altitude:	38 words or 152 bytes per record* (IRECLS)
Global climatology:	39 words or 156 bytes per record (IRECLG)

* To conserve disk space, it is recommended that the global scene/altitude data base be installed using INTEGER*2 variables. If you choose to do so, modify the record length appropriately for 19 words or 76 bytes per record. Other modifications will be discussed in INSTDB and MOSART.

Known conventions for different computers are as follows:

VAX:	Number of 4-byte words
SGI:	Number of 4-byte words
Primos:	Number of 2-byte words
Sun:	Number of bytes
PC Lahey:	Number of bytes
Hewlett-Packard:	Number of bytes
Data General:	Number of bytes
IBM VM/CMS:	Number of bytes + 4 bytes padding

If your computer is different from the VAX and SGI convention, you will need to modify the variables IRECL (all of them), IRECLS, and IRECLG.

FPTEST will determine the record lengths for the three file sizes used in MOSART and will provide these values to the user.

3.6.5 I/O Control Switches

The vector ICMPTR is used to control the input and output of the MOSART program.

ICMPTR(1) is a switch to control the OPEN commands for different computers. ICMPTR(1)=0 implies that only one version of a file is retained, as on a Unix machine. ICMPTR(1)=1 implies that different versions of a file are retained, as on a VAX machine. The MOSART program and its utility programs use the STATUS='OLD' or STATUS='NEW' parameter in OPEN statements to distinguish between old and new version of a file.

ICMPTR(2) is a switch for the existence of column 1 in FORTRAN output. Some computers handle the carriage controls (i.e., the characters in column 1 in a FORTRAN output file) as simply another character in the file. For these computers, the character is visible when viewed by an editor and printed on a printer unless the printer command dictates that they be hidden and used as a carriage control (e.g., the lpr -f or as a filter in Unix). Other computers remove the first column from the file. The two methods make no difference to the MOSART program, but several of the utility programs create files for which the occurrence or nonoccurrence of the first column is critical (e.g., in the input file creator program). ICMPTR(2)=0 implies that column 1 appears in all FORTRAN ASCII output files, while ICMPTR(2)=1 implies that it is suppressed.

3.6.6 Maximum Record Length

The Sun computer sets a maximum record length for binary files of 2045 words. To accommodate this limitation, the variable LRMAX is defined as the maximum record length. As delivered, it is set to 1024. If your computer requires a shorter maximum record length, simply redefine this variable.

3.7 CHARACTER*72 FUNCTION IOERR

To assist the user in determining any I/O messages received from MOSART, two (2) methods are used. First, the input/output status specifier IOSTAT is used on all Input/Output statements (e.g., OPEN, INQUIRE, REWIND, READ, WRITE, and CLOSE) with the exception of simple WRITE statements to the terminal. If an I/O error occurs, an error message is written to the terminal with a brief explanation of where the error occurred, perhaps some auxiliary information, and the value of IOSTAT.

According to the ANSI X3.9-1978 Standard, "execution of an input/output statement containing this specifier [i.e., IOSTAT=ios] causes ios to become defined:

(1) with a zero value if neither an error condition nor an end-of-file condition is encountered by the processor,

(2) with a processor dependent positive integer value if an error condition is encountered, or

(3) with a processor-dependent negative integer value if an end-of-file condition is encountered and no error condition is encountered."

To assist the MOSART user, the FUNCTION IOERR is used to translate the value ios into a text message, which is printed along with the value of ios. Since any error or end-of-file condition will return a processor dependent value, IOERR must be processor dependent. Currently, IOERR is coded for Unix, VAX, and PC Lahey F77L and F77L EM/32 processors, as well as for a generic message based upon whether ios is zero, positive, or negative. Each of these implementations is discussed briefly below. If your system uses a different method for providing the I/O error message, please contact the author so that it can be included in future releases of MOSART.

For Unix systems (e.g., Silicon Graphics, Hewlett-Packard), the CHARACTER FUNCTION GERROR() is used. GERROR returns the system error message in a CHARACTER variable string. The length of the string returned by GERROR is determined by the calling program. As implemented in IOERR, it is called as a CHARACTER*72 FUNCTION.

For the VAX system, the use of IOSTAT=IOS disables the printing of error messages to the terminal. Therefore, the VAX I/O error messages are hard-wired into IOERR. To implement them, simply replace the string 'CVAX' with blanks after commenting out all references to GERROR with the string 'CUNX'.

For the PC Lahey compilers, the Lahey-provided IOSTAT_MSG subroutine is used. Since IOSTAT_MSG returns a message of 152 characters, it is truncated at 72 for use by IOERR. IOSTAT_MSG is passed the value IOS, and assigns a message text for the error detected by the input/output statement. If the message cannot be read because no file handles are available, the error file is not on the path, or the message number is invalid, then the message returned in blank. To implement IOSTAT_MSG, simply replace the string 'CLAH' with blanks after commenting out all references to GERROR with the string 'CUNX'.

For the generic system, simply replace the string 'CGEN' with blanks after commenting out all references to GERROR with the string 'CUNX'.

3.8 Compiling, Loading, and Executing

Once all the required modifications have been made to the FPTEST program, it should be compiled, loaded, and executed as an ANSI X3.9-1978 FORTRAN program. The FPTEST program requires a response to the testing of the SUBROUTINE PROMPT at the end of the output (see below). Any character plus a carriage return is acceptable. FPTEST provides all output to the screen, unless such output is redirected by the user.

3.9 Sample Output

The FPTEST program is delivered for immediate use on a Silicon Graphics computer. For the Silicon Graphics computer, the output from the FPTEST program is shown in Table 22.

Table 22. Silicon Graphics FPTEST Output.

```
MOSART Radiative Environment Summary (Ver. 1.40) Wed Nov  3 14:39:34 1993
Floating Point Number Test

Radix                          =           2
Big-endian representation
Floating-point addition with IEEE rounding
Gradual underflow
Two's complement representation
Register and storage calculations have the same precision
No. of guard digits            =           0

                                --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand     =       24              53
No. of bits in exponent        =        8              11
Smallest exponent for 1+e      =      -23             -52
Smallest exponent for 1-e      =      -24             -53
Maximum exponent               =      128            1024
Minimum exponent               =     -126           -1022
Pi (3.14159265358979323846264) =      3.141592741      3.14159265358979310
e  (2.71828182845904523536028) =      2.718281746      2.71828182845904500
Small Positive number          =  1.175494351E-38  2.22507385850720220E-308
Large Positive number          =  3.402823466E+38  1.79769313486231500E+308
Small Difference number (1+e)  =  1.192092896E-07  2.22044604925031300E-016
Small Difference number (1-e)  =  5.960464478E-08  1.11022302462515650E-016
Smallest denormalized number   =  1.401298464E-45  4.94065645841246850E-324

The printing will place meaningful information starting in column 1.

All file handling assumes that only the latest version of a file is saved.

This computer does not initialize variables.

This computer operates in dynamic mode.

Direct access file record lengths:         Calculated         Stored Value
  Molecular Data Bases:                         48                 48
  Scene Data Base (INTEGER*2):                  19                 38
  Global Data Base:                             39                 39

If stored values differ from calculated values, please change.

Is the carriage control suppressed (Y/N):

STOP Normal termination statement executed
```

For an IBM PC using the Lahey F77L compiler, the time and date calls must be changed in SUBROUTINE TITLCR.

Also, in BLOCK DATA DEVCBD, the record length must be changed to number of bytes. In SUBROUTINE CONFIG, the call to UNDLF0 must be included if a 286 machine is used. If these changes are made, the results are shown in Tables 23 and 24.

Table 23. Lahey (Version 5.0) FPTEST Output.*

```
MOSART Radiative Environment Summary (Ver. 1.00) 10:27:40.43 02/26/93
Floating Point Number Test

Radix                              =             2
Little-endian representation
Floating-point addition with IEEE rounding
Abrupt underflow
Two's complement representation
Register and storage calculations have different precisions
No. of guard digits                =             0

                                      --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand         =         24              53
No. of bits in exponent            =          8              11
Smallest exponent for 1+e          =        -23             -52
Smallest exponent for 1-e          =        -24             -53
Maximum exponent                   =        128            1024
Minimum exponent                   =       -126           -1022
Pi (3.14159265358979323846264) =      3.14159        3.14159265358979
e  (2.71828182845904523536603) =      2.71828        2.71828182845905
Small Positive number              =   0.117549E-37   0.222507385850720D-307
Large Positive number              =   0.340282E+39   0.179769313486232D+309
Small Difference number (1+e)  =   0.119209E-06   0.222044604925031D-015
Small Difference number (1-e)  =   0.596046E-07   0.111022302462516D-015
Smallest denormalized number   =   0.117549E-37   0.222507385850720D-307


The printing will place meaningful information starting in column 1.

All file handling assumes that only the latest version of a file is saved.

Direct access test successfully completed.

Is the carriage return suppressed (Y/N):
```

* Direct access file record lengths are not included in this table, but should appear in output

Table 24. Lahey (Version 5.1) FPTEST Output.

```
MOSART Radiative Environment Summary (Ver. 1.00) 10:33:20 09/28/93
Floating Point Number Test

Radix                              =          2
Little-endian representation
Floating-point addition with IEEE rounding
Gradual underflow
Two's complement representation
Register and storage calculations have different precisions
No. of guard digits                =          0

                                    --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand         =         24           53
No. of bits in exponent            =          8           11
Smallest exponent for 1+e          =        -23          -52
Smallest exponent for 1-e          =        -24          -53
Maximum exponent                   =        128         1024
Minimum exponent                   =       -126        -1022
Pi (3.14159265358979323846264) =    3.14159           3.14159265358979
e  (2.71828182845904523536028) =    2.71828           2.71828182845905
Small Positive number              =   0.117549E-37  0.222507385850720D-307
Large Positive number              =   0.340282E+39  0.179769313486232D+309
Small Difference number (1+e) =   0.119209E-06  0.222044604925031D-015
Small Difference number (1-e) =   0.596046E-07  0.111022302462516D-015
Smallest denormalized number       =   0.140130E-44  0.494065645841246D-323


The printing will place meaningful information starting in column 1.


All file handling assumes that only the latest version of a file is saved.


Direct access file record lengths:     Calculated         Stored Value
 Molecular Data Bases:                    192                  48
 Scene Data Base (INTEGER*2):              76                  38
 Global Data Base:                        156                  39


If stored values differ from calculated values, please change.


Is the carriage control suppressed (Y/N):
```

If the direct access record length is not changed, a difference between calculated and stored values is presented. The major difference between Lahey PC outputs for Versions 5.0 and 5.1 is that Version 5.1 implements IEEE gradual underflow with a smallest denormalized number significantly smaller than the smallest positive number. Version 5.0 implements abrupt underflow, so the two numbers are the same. The Lahey PC implementation has more precision in its register calculations. Also, the Lahey PC uses little-endian representation.

FPTEST output for an HP9000 computer is shown in Table 25 and for a Data General in Table 26.

Table 25. HP9000 FPTEST Output.

```
MOSART Radiative Environment Summary (Ver. 1.25) Fri Sep 23 13:47:07 1994
Floating Point Number Test

Radix                             =  2
Big-endian representation
Floating-point addition with IEEE rounding
Gradual underflow
Two's complement representation
Register and storage calculations have the same precision
No. of guard digits               =  0

                                     --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand        =       24                 53
No. of bits in exponent           =        8                 11
Smallest exponent for 1+e         =      -23                -52
Smallest exponent for 1-e         =      -24                -53
Maximum exponent                  =      128               1024
Minimum exponent                  =     -126              -1022
Pi (3.14159265358979323846264) =      3.141592741      3.14159265358979320
e  (2.71828182845904523536028) =      2.718281746      2.71828182845904490
Small Positive number             =  1.175494351E-38  2.22507385850720280E-308
Large Positive number             =  3.402823466E+38  1.79769313486231500E+308
Small Difference number (1+e) =  1.192092896E-07  2.22044604925031300E-016
Small Difference number (1-e) =  5.960464478E-08  1.11022302462515640E-016
Smallest denormalized number  =  1.401298464E-45  4.94065645841246640E-324

The printing will place meaningful information starting in column 1.

All file handling assumes that only the latest version of a file is saved.

This computer does not initialize variables.

This computer operates in dynamic mode.

Direct access file record lengths:        Calculated          Stored Value
  Molecular Data Bases:                       192                 192
  Scene Data Base (INTEGER*2):                 76                  76
  Global Data Base:                           156                 156
  UFTAPE Data Base:                         13000               13000

If stored values differ from calculated values, please change.

Is the carriage control suppressed (Y/N):
```

Table 26. Data General FPTEST Output.*

```
MOSART Radiative Environment Summary (Ver. 1.00) 10:27:40.43 02/26/93
Floating Point Number Test

Radix                            =            2
Little-endian representation
Floating-point addition with IEEE rounding
Abrupt underflow
Two's complement representation
Register and storage calculations have different precisions
No. of guard digits              =            0

                                       --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand       =           24            53
No. of bits in exponent          =            8            11
Smallest exponent for 1+e        =          -23           -52
Smallest exponent for 1-e        =          -24           -53
Maximum exponent                 =          128          1024
Minimum exponent                 =         -126         -1022
Pi (3.14159265358979323846 26) =      3.141593          3.14159265358979
e  (2.71828182845904523536 03) =      2.718282          2.71828182845904
Small Positive number            =     1.1754944E-38   2.2250738585072022-308
Large Positive number            =     3.4028235E+38   1.7976931348623150+308
Small Difference number (1+e)    =     1.1920929E-07   2.2204460492503130E-16
Small Difference number (1-e)    =     5.9604645E-08   1.1102230246251565E-16
Smallest denormalized number     =     1.1754944E-38   2.2250738585072022-308

The printing will place meaningful information starting in column 1.


All file handling assumes that only the latest version of a file is saved.


Direct access test successfully completed.


Is the carriage control suppressed (Y/N):


(Note:  Carriage control was not suppressed.)
```

* Direct access file record lengths are not included in this table, but should appear in output

On a VAX using D-float representation, the small and large DOUBLE PRECISION numbers will be approximately the same as the REAL numbers; in G-float representation, the values will be similar to those given above, as shown in Tables 27 and 28, respectively. For the VAX, ICMPTR(1) must be set to 1, and ICMPTR(2) must be set to 0.

The output for a Sun is very similar to that for the Silicon Graphics (see Table 29). However, the record lengths for all direct access files need to be changed to the number of bytes. For Digital Alpha DEC 3000 machines, the results are given in Table 30.

Table 27. VAX FPTEST Output D-Float Representation.

```
MOSART Radiative Environment Summary (Ver. 1.00) 17:04:27 15-JUL-93
Floating Point Number Test

Radix                          =           2
Little-endian representation
Floating-point addition with non-IEEE rounding
Abrupt underflow
Two's complement representation
Register and storage calculations have the same precision
No. of guard digits            =           0

                                   --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand     =          24              56
No. of bits in exponent        =           8               8
Smallest exponent for 1+e      =         -24             -56
Smallest exponent for 1-e      =         -24             -56
Maximum exponent               =         127             127
Minimum exponent               =        -128            -128
Pi (3.14159265358979323846264) =    3.141593        3.141592653589793
e  (2.71828182845904523536030) =    2.718282        2.718281828459045
Small Positive number          =   2.9387359E-39  2.9387358770557188E-39
Large Positive number          =   1.7014117E+38  1.7014118346046923E+38
Small Difference number (1+e)  =   5.9604645E-08  1.3877787807814457E-17
Small Difference number (1-e)  =   5.9604645E-08  1.3877787807814457E-17
Smallest denormalized number   =   2.9387359E-39  2.9387358770557188E-39


The printing will place meaningful information starting in column 1.


All file handling assumes that multiple versions of a file are saved.

Direct access file record lengths:     Calculated          Stored Value
  Molecular Data Bases:                    48                   48
  Scene Data Base (INTEGER*2):             19                   38
  Global Data Base:                        39                   39


If stored values differ from calculated values, please change.


Is the carriage control suppressed (Y/N):
```

Table 28. VAX FPTEST Output G-Float Representation.

```
MOSART Radiative Environment Summary (Ver. 1.00) 17:04:27 15-JUL-93
Floating Point Number Test

Radix                             =            2
Little-endian representation
Floating-point addition with non-IEEE rounding
Abrupt underflow
Two's complement representation
Register and storage calculations have the same precision
No. of guard digits               =            0

                                    --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand        =           24             53
No. of bits in exponent           =            8             11
Smallest exponent for 1+e         =          -24            -53
Smallest exponent for 1-e         =          -24            -53
Maximum exponent                  =          127           1023
Minimum exponent                  =         -128          -1024
Pi (3.14159265358979323846 26) =     3.141593        3.14159265358979
e  (2.71828182845904523536 03) =     2.718282        2.71828182845905
Small Positive number             =    2.9387359E-39  5.562684646268003E-309
Large Positive number             =    1.7014117E+38  8.988465674311579E+307
Small Difference number (1+e)     =    5.9604645E-08  1.110223024625157E-016
Small Difference number (1-e)     =    5.9604645E-08  1.110223024625157E-016
Smallest denormalized number      =    2.9387359E-39  5.562684646268003E-309


The printing will place meaningful information starting in column 1.


All file handling assumes that multiple versions of a file are saved.

Direct access file record lengths:      Calculated         Stored Value
  Molecular Data Bases:                     48                  48
  Scene Data Base (INTEGER*2):              19                  38
  Global Data Base:                         39                  39


If stored values differ from calculated values, please change.


Is the carriage control suppressed (Y/N):
```

Table 29. Sun FPTEST Output.

```
MOSART Radiative Environment Summary (Ver. 1.25) Mon Oct 17 11:42:42 1994
Floating Point Number Test

Radix                         =   2
Big-endian representation
Floating-point addition with IEEE rounding
Gradual underflow
Two's complement representation
Register and storage calculations have the same precision
No. of guard digits           =   0

                                  --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand    =        24                 53
No. of bits in exponent       =         8                 11
Smallest exponent for 1+e     =       -23                -52
Smallest exponent for 1-e     =       -24                -53
Maximum exponent              =       128               1024
Minimum exponent              =      -126              -1022
Pi (3.1415926535897932384626) =     3.141592741       3.14159265358979312
e  (2.7182818284590452353603) =     2.718281746       2.71828182845904553
Small Positive number         =  1.175494351E-38  2.22507385850720138E-308
Large Positive number         =  3.402823466E+38  1.79769313486231571E+308
Small Difference number (1+e) =  1.192092896E-07  2.22044604925031308E-016
Small Difference number (1-e) =  5.960464478E-08  1.11022302462515654E-016
Smallest denormalized number  =  1.401298464E-45  4.94065645841246544E-324

The printing will place meaningful information starting in column 1.

All file handling assumes that only the latest version of a file is saved.

This computer MAY initialize variables to zero.

This computer operates in dynamic mode.

Direct access file record lengths:         Calculated          Stored Value
  Molecular Data Bases:                       192                   48
  Scene Data Base (INTEGER*2):                 76                   19
  Global Data Base:                           156                   39
  UFTAPE Data Base:                         13000                 3250

If stored values differ from calculated values, please change.

Is the carriage control suppressed (Y/N):  Y
STOP: Normal termination
```

Table 30. Digital Alpha DEC 3000 Output.

```
MOSART Radiative Environment Summary (Ver. 1.25) Mon Aug  8 11:50:04 1994
Floating Point Number Test

Radix                            =             2
Little-endian representation
Floating-point addition with IEEE rounding
Abrupt underflow
Two's complement representation
Register and storage calculations have the same precision
No. of guard digits              =             0

                                    --- REAL --- --- DOUBLE PRECISION ---
No. of bits in significand       =       24                 53
No. of bits in exponent          =        8                 11
Smallest exponent for 1+e        =      -23                -52
Smallest exponent for 1-e        =      -24                -53
Maximum exponent                 =      128               1024
Minimum exponent                 =     -126              -1022
Pi (3.14159265358979323846264) =       3.141592741       3.14159265358979312
e  (2.71828182845904523536030) =       2.718281746       2.71828182845904509
Small Positive number            =   1.175494351E-38  2.22507385850720138E-308
Large Positive number            =   3.402823466E+38  1.79769313486231571E+308
Small Difference number (1+e) =   1.192092896E-07  2.22044604925031308E-016
Small Difference number (1-e) =   5.960464478E-08  1.11022302462515654E-016
Smallest denormalized number     =   1.175494351E-38  2.22507385850720138E-308

The printing will place meaningful information starting in column 1.

All file handling assumes that only the latest version of a file is saved.

This computer MAY initialize variables to zero.

This computer operates in dynamic mode.

Direct access file record lengths:          Calculated          Stored Value
  Molecular Data Bases:                         48                   48
  Scene Data Base (INTEGER*2):                  19                   19
  Global Data Base:                             39                   39
  UFTAPE Data Base:                           3250                 3250

If stored values differ from calculated values, please change.

Is the carriage control suppressed (Y/N):
```

These sample cases may not agree exactly with the output from your machine. If you do not get reasonable numbers (e.g., NaN or Inf on a machine conforming to the IEEE Standard 754, or ridiculous numbers such a 0.0 for smallest positive number), either determine where the code is producing the problem and fix it, or hard-wire the proper values in the program. In either case, please contact the author.

> Note: Several of the outputs shown used early versions of FPTEST. Some new features (e.g., direct access record lengths, variable initialization, static vs. dynamic mode) are not shown.

4.0 INSTALLING THE DATA BASES

Once you have successfully completed modifying the various routines in FPTEST (and making the identical modifications to all the other programs that contain these routines if the code was delivered in a bundled format), you must install the data bases. All the data bases are delivered as sequential access ASCII data files. With the exception of the SAG/NRL data base, all files must be converted to direct access binary data files.

The easiest way to do this is to read in the data bases from the delivered tape and name each file as follows:

file01.dat	CH4.NRL
file02.dat	CO.NRL
file03.dat	H2O.NRL
file04.dat	HNO3.NRL
file05.dat	N2O.NRL
file06.dat	NO2.NRL
file07.dat	O3D.NRL
file08.dat	O3R.NRL
file09.dat	OAT.NRL
file10.dat	TEMP.NRL
file11.dat	
file12.dat	
UFTAPE.ASC	
global.dat	
scenes.dat	

If you have modified the BLOCK DATA DEVCBD as discussed above, all you should have to do is compile INSTDB, load it, and execute it. For example, on a Unix machine, type the following:

```
f77 -C -g -u -o instdb instdb.f
instdb
```

For other machines, different commands are required.

Installation Clarification. When installing the MOSART data bases, if the script utility direc.make is used to create a directory structure and place the files in the appropriate directories, it is important to run INSTDB from the appropriate location (particularly if default sequential access ASCII file names are used). Using the script auto.run takes care of this for you; however, if you wish to execute INSTDB manually, you must do so from either the main mosart directory (i.e., the directory in which you unloaded the tape) or the directory where the sequential access ASCII data files are located. If you used the direc.make utility to create thhe directory structure, the sequential access ASCII data files are located in data/ascii/.

As discussed in the installation of FPTEST, the IBM VM/CMS system requires additional information before OPENing a direct access file. The IBM routine required in INSTDB is commented out by 'CIBM'. If you are installing this program on an IBM computer, you will need to replace this string with blanks, just as you did in FPTEST.

With the release of Version 1.40, the size of the scene data base (i.e., ecosdb.dat) has grown considerably. Although the contents of the direct access binary data file can be created (by INSTDB) as a set of INTEGERs, for size reasons, it is highly desirable to pack the contents of three INTEGERs into a single 4-byte word. Each 4-byte word would then contain:

Bits 0 - 7: Ecosystem index (0-77)
Bits 8 - 15: Percent water (0-100)
Bits 16 - 31: Altitude (m) above mean sea level (~400 to 8500)

One of the following methods should be used:

Method 1: If INTEGER*1 and INTEGER*2 are available on your compiler, declare the variables JBK and ITWR as INTEGER*1. IALT should be declared as INTEGER*2. This is the configuration of the code as delivered, so no changes are required to the code.

Method 2: If INTEGER*1 is not available on your computer, but the MIL-STD function IBITS is, use the code commented out by "CMIL" in INSTDB and RDSCN to pack and unpack the code. (Note: for VAX and Lahey compilers, the function is called JIBITS and is commented out by "CVAX" and "CLAH", respectively.)

Method 3: If INTEGER*1 and IBITS are not available on your computer, but INTEGER*2 is, declare JBK and IWTR as INTEGER*2. This code is commented out by "CINT2" in RDSCN and INSTDB. Also, you should use the code commented out by "CINT2" in FPTEST to determine the record length required in BLOCK DATA DEVCBD.

Method 4: If none of the above options are available, declare JBK, IWTR, and IALT as INTEGER in RDSCN and INSTDB (commented out by "CINT"). Similar changes to Method 3 need to be made in FPTEST and BLOCK DATA DEVCBD.

Upon execution, you will be asked if you want to use the default names. If you named the sequential access ASCII files as shown above and want to used the names for the direct access binary files that are in BLOCK DATA DEVCBD, simply respond 'Y' for Yes. You will then be informed as each file is installed what molecule the file is for, how many records it has, and a check sum as shown below:

```
Use default file names (Y/N):

Conversion completed for Water vapor
        /proj/mosart/data/h2oxx.dat
         19468 Records with the check sum =     10166916
```

Conversion completed for Carbon dioxide
/proj/mosart/data/co2xx.dat
10005 Records with the check sum = 3707730

Conversion completed for Ozone
/proj/mosart/data/o3xxx.dat
3049 Records with the check sum = 1980916

Conversion completed for Nitrous oxide
/proj/mosart/data/n2oxx.dat
3567 Records with the check sum = 2200572

Conversion completed for Carbon monoxide
/proj/mosart/data/coxxx.dat
2475 Records with the check sum = 800572

Conversion completed for Methane
/proj/mosart/data/ch4xx.dat
4760 Records with the check sum = 2310657

Conversion completed for Oxygen
/proj/mosart/data/o2xxx.dat
2638 Records with the check sum = 455051

Conversion completed for Nitric oxide
/proj/mosart/data/noxxx.dat
1524 Records with the check sum = 581680

Conversion completed for Sulfur dioxide
/proj/mosart/data/so2xx.dat
1042 Records with the check sum = 1087970

Conversion completed for Nitrogen dioxide
/proj/mosart/data/no2xx.dat
1095 Records with the check sum = 623446

Conversion completed for Ammonia
/proj/mosart/data/nh3xx.dat
2184 Records with the check sum = 1327014

Conversion completed for Nitric acid
/proj/mosart/data/hno3x.dat
1146 Records with the check sum = 930885

Conversion completed for UFTAPE file
273 Records with the check sum = 26511562

```
Conversion completed for Global Data Base
        /long/mosart/data/global.dat
         24840 Records with the check sum =     7155602

Conversion completed for Scene Data Base
        /long/mosart/data/scenes.dat
         64800 Records with the check sum =      783011

STOP Normal termination statement executed
```

If for some reason, you do not wish to use the default names, respond 'N' for No (any response other than 'Y' or 'y' is equivalent to No). You will then be prompted for each sequential access ASCII file and the corresponding direct access binary file.

If your output does not agree with the number of records and check sums given above, please contact the author.

Version 1.40 of the MOSART code can operate with different sets of molecular data bases:

1. The twelve MOSART data bases installed above,
2. The UFTAPE data base used (and installed with) MODTRAN,
3. Any combination of MOSART data bases and the UFTAPE data base.

If you do not have MODTRAN installed on your system, make no changes to the MOSART data bases. If you do have MODTRAN installed and have placed the directory and file name for the UFTAPE data base in BLOCK DATA DEVCBD, you can delete all twelve (12) MOSART molecular data bases (but not the global or scene/altitude data bases); however, it is recommended that you back up these data bases before you delete them.

The structure of the MOSART molecular data bases allows the user to remove (or add) molecules as required. The structure of the UFTAPE data base requires all twelve (12) molecules to be included in the calculation. The MOSART code, therefore, is also designed to use any MOSART data base files that exist, and will supplement the molecular band parameters with the values from the UFTAPE data base only if the file name in BLOCK DATA DEVCBD does not exist.

To conserve disk space, the user can delete (preferably after being backed up) either the global data file or the scene/altitude data file (or both). However, if the MOSART code cannot find either of these files, it uses default values (e.g., sea level altitude, ocean background, model atmosphere, surface air temperature, no cloud cover).

Using the Unix environment. Several users requested a method for initializing the data base directory path (the variable DIRPTH in BLOCK DATA DEVCBD) at run time, rather than at compile time. This has been accomplished using the setenv command in Unix, where

setenv MOSART_DIRPTH 'data_base_directory'

is placed in the .cshrc file. In the main program, the declaration and call to GETENV should be uncommented, along with the line following the call to GETENV. With this routine, the user can move the data bases to different directories without having to recompile the code. If there is no entry in .chsrc for MOSART_DIRPTH, then the value in BLOCK DATA DEVCBD is used.

Unfortunately, this will only work on Unix machines. If anyone is aware of a technique that will work on other computers (e.g., VAX VMS, PC), please contact Dr. William M. Cornette, (619) 455-9741 or wmc@photon.com.

5.0 INSTALLING MOSART

Once you have made any changes in the MOSART program routines CONFIG, CNSTNT, TITLCR, IOERR, and DEVCBD that were discussed in Section 3.0, and once you have installed the data bases (Section 4.0), you can compile and load the MOSART program. However, there are a couple of changes that you may wish to make:

1. You should have already changed PROGRAM INSTDB, SUBROUTINE RDSCN, and BLOCK DATA DEVCBD as discussed in Section 4.0, with regard to packing the contents of the scene data base.

2. If your computer supports either COMPLEX*16 or DOUBLE COMPLEX, it is recommended that you make the indicated changes to the declarations in SUBROUTINEs MIE and COAT. For some computers, you may also have to change the generic INTRINSIC function AIMAG to DIMAG.

3. If supported on your computer (e.g., on a VAX or Lahey PC), you may want to change the OPENing of the data base files in SUBROUTINEs DBINIT, RDSCN, and RDGBL to be in a read-only mode. If it is possible to OPEN these files for multiple access, these changes should also be made.

4. On an IBM VM/CMS system, you must uncomment the calls to SUBROUTINE FILEINF in SUBROUTINEs DBINIT, RDSCN, and RDGBL, just as you did in FPTEST and INSTDB.

5. On an IBM VM/CMS system, you must uncomment the lines in SUBROUTINE FILRT for placing '/ ' before the file name, as discussed above.

6. In SUBROUTINE ABSMOL, a mixed mode access to the molecular data bases is incorporated. The first READ of a data base is direct access, while subsequent READs are sequential access. This mixed mode reduces the execution time; however, if mixed mode READs are not permitted on your machine, comment out the portion of code as noted in this module.

7. <u>C++ Wrapper</u>. A C++ wrapper is provided for those users who desire that the MOSART code be incorporated into a C++ environment. If the C++ wrapper (in a file called wrapper.c) is to be used, then PROGRAM MOSART must be modified as follows:

    1. The statement "PROGRAM MOSART" should be commented out and the second line (SUBROUTINE MOSART(FILERT)) should be used instead;

    2. The declaration for FILERT should be changed as noted in the code; and

    3. The CALL PROMPT ("File root name: ") and the subsequent READ statement should be commented out.

8. Clearing Flags on a Sun Computer. The Sun computer will produce warning messages at the end of a job with regard to Inexact and Underflow occurrences, even though these are expected during the execution of MOSART. If the user wishes to avoid these warnings, in the main program replace the occurrences of "CSUN" with blanks so that the routine CLEAR is called, and compile and load the file clear.c with the rest of the code. It should be noted that this routine will not remove warning messages concerning overflows and illegal operations.

5.1 Bundled Delivery

If you have the bundled version of MOSART, compile, link, and load MOSAR1.f, MOSAR2.f, MOSAR3.f, MOSBD1.f, MOSBD2.f, and MACHINE.f.

5.2 Unbundled Delivery

For a Unix machine, use make with the file Make_mosart. Otherwise compile, link, and load all source codes in the directories src/ and src/BD.

5.3 Execution

A test input file, test.in, is supplied with the program (see Appendix A). For comparison, a test output is also provided. Do not name the delivered test output "test.out" since this file is over-written by the MOSART program. This test program requires approximately 18 seconds of CPU time on a Silicon Graphics Personal IRIS 4D/35, operating at 36 MHz.

The following information should be output to the terminal (or device '*'):

```
        M       M   OOOOO     SSSSS       A     RRRRRRR   TTTTTTTTT
        MM     MM  O     O   S     S     A A    R      R      T
        M M   M M O       O S       S   A   A   R       R     T
        M  M M  M O       O S          A     A  R       R     T
        M   M   M O       O  S        A       A R      R      T
        M       M O       O   SSSSS   AAAAAAAAA RRRRRRR       T
        M       M O       O        S  A       A R   R         T
        M       M O       O         S A       A R    R        T
        M       M O       O S       S A       A R     R       T
        M       M  O     OO  S     S  A       A R      R      T
        M       M   OOOOO     SSSSS   A       A R       R     T

                              (Version 1.30)
                             (26 October 1994)

File root name:

    1:  User-specified Parameters ------------------------
    2:  Position Parameters ------------------------------
    3:  Geometry Parameters ------------------------------
    4:  Spectral Parameters ------------------------------
```

```
Warning No. 21:  Source altitude less than terrain altitude.
                 Source altitude no.     2 set to       0.243 km.

Warning No. 21:  Source altitude less than terrain altitude.
                 Source altitude no.     4 set to       0.243 km.

Sample Input File
MOSART Radiative Environment Summary (Ver. 1.30) Wed Nov  3 12:59:13 1994
Scratch file number     29 OPENed.  Type:  Geometry Parameters
```

Geometry Conditions (12 positions)

No		Observer Altitude (km)	Src/Tang Altitude (km)	Slant Range (km)	Earth Center Angle (deg)	Obs. Look Angle (deg)	Src. Look Angle (deg)
1	Se	100.00	1.00	1049.99	9.35	-10.01	1.00
2	Be	100.00	0.24	99.77	0.00	-90.00	90.00
3	Ce	100.00	1.00	99.01	0.00	-90.00	90.00
4	Ce	100.00	88.56	120.00	1.06	-6.00	4.94
5	Cz	100.00	1.00	99.01	0.00	180.00	0.00
6	Cl	100.00	1.00	500.56	4.39	-13.58	9.27
					Lat:	46.00	Lon: -98.20
7	Ae	1.00					
8	He	1.00		1.00	0.01		
9	Le	400.00	349.57			-7.00	
10	Le	100.00	1.00			-9.96	
11	Le	400.00	-200.00			-24.28	
12	Le	400.00	0.38			-19.75	

```
STOP Normal Termination statement executed
```

Some of the features may differ slightly for your computer (e.g., the STOP statement), but the numerical results should be essentially the same.

6.0 INSTALLING THE MOSART UTILITIES

6.1 ASCBIN

6.1.1 Bundled Delivery

Once you have made the necessary changes to the SUBROUTINEs PROMPT and CONFIG, the FUNCTION IOERR, and the BLOCK DATA DEVCBD, simply compile the ASCBIN.f file, and link and load it with MACHIN.o.

6.1.2 Unbundled Delivery

For a Unix machine, use make with the file Make_ascbin. Otherwise, compile the files in the directory ascbin_src/ and link and load with appropriate object files.

6.2 BBTEMP

6.2.1 Bundled Delivery

Once you have made the necessary changes to the SUBROUTINEs CNSTNT, PROMPT, and CONFIG, the FUNCTION IOERR, and the BLOCK DATA DEVCBD, simply compile the BBTEMP.f file, and link and load it with MACHIN.o.

6.2.2 Unbundled Delivery

For a Unix machine, use make with the file Make_bbtemp. Otherwise, compile the files in the directory bbtemp_src/ and link and load with appropriate object files.

6.3 CRFILE

6.3.1 Bundled Delivery

Once you have made the necessary changes to the SUBROUTINEs PROMPT and CONFIG, the FUNCTION IOERR, and the BLOCK DATA DEVCBD, simply compile the CRFILE.f file, and link and load it with MACHIN.o.

6.3.2 Unbundled Delivery

For a Unix machine, use make with the file Make_crfile. Otherwise, compile the files in the directory crfile_src/ and link and load with appropriate object files.

6.4 FACET

6.4.1 Bundled Delivery

Once you have made the necessary changes to the SUBROUTINEs CNSTNT, PROMPT, and CONFIG, the FUNCTION IOERR, simply compile the FACET.f file, and link and load it with MACHIN.o.

6.4.2 Unbundled Delivery

For a Unix machine, use make with the file Make_visual. Otherwise, compile the files in the directory facet_src/ and link and load with appropriate object files.

6.5 MRFLTR

6.5.1 Bundled Delivery

Once you have made the necessary changes to the SUBROUTINEs CNSTNT, PROMPT, and CONFIG, the FUNCTION IOERR, and the BLOCK DATA DEVCBD, simply compile the MRFLTR.f file, and link and load it with MACHIN.o.

6.5.2 Unbundled Delivery

For a Unix machine, use make with the file Make_mrfltr. Otherwise, compile the files in the directory mrfltr_src/ and link and load with appropriate object files.

6.6 PLTGEN

6.6.1 Bundled Delivery

If you have the NCAR plotting package, once you have made the necessary changes to the SUBROUTINEs PROMPT and CONFIG, and the FUNCTION IOERR, simply compile the PLTGEN.f file. It can then be linked and loaded with MACHIN.o, referencing the appropriate libraries for the NCAR plotting package. The main PROGRAM PLTGEN also contains a call to the SUBROUTINE SYSTEM, which executes a command file for submitting the plotter file to a printer. Please either modify this line for your computer or comment it out.

6.6.2 Unbundled Delivery

For a Unix machine, use make with the file Make_pltgen. Otherwise, compile the files in the directory pltgen_src/ and link and load with appropriate object files and the NCAR library.

6.7 SCNGEN

6.7.1 Bundled Delivery

Once you have made the necessary changes to the SUBROUTINEs CNSTNT, PROMPT, and CONFIG, and the FUNCTION IOERR, simply compile the SCNGEN.f file, and link and load it with MACHIN.o.

6.7.2 Unbundled Delivery

For a Unix machine, use make with the file Make_scngen. Otherwise, compile the files in the directory scngen_src/ and link and load with appropriate object files.

6.8 TERTEM

6.8.1 Bundled Delivery

Once you have made the necessary changes to the SUBROUTINEs CNSTNT, PROMPT, and CONFIG, and the FUNCTION IOERR, simply compile the TERTEM.f file, and link and load it with MACHIN.o.

6.8.2 Unbundled Delivery

For a Unix machine, use make with the file Make_tertem. Otherwise, compile the files in the directory tertem_src/ and link and load with appropriate object files.

6.9 VISUAL

6.9.1 Bundled Delivery

Once you have made the necessary changes to the SUBROUTINEs CNSTNT, PROMPT, and CONFIG, and the FUNCTION IOERR, simply compile the VISUAL.f file, and link and load it with MACHIN.o.

6.9.2 Unbundled Delivery

For a Unix machine, use make with the file Make_visual. Otherwise, compile the files in the directory visual_src/ and link and load with appropriate object files.

7.0 CLEAN-UP

A number of files have been created that are not required after installation. FPTEST and INSTDB can be deleted and the ASCII data files (located in data/ascii/.) can be compressed, or if you wish to save space, these files can be deleted. Further space can be saved by the following:

1. If you placed the file name of the MODTRAN UFTAPE in BLOCK DATA DEVCBD, you can delete all of the binary molecular data bases.

2. If you wish to use default values, you can delete either the global data base or the scene data base (or both).

8.0 MOSART INPUT BUILDER

The MOSART Input Builder (MIB) is a graphical user interface that allows the user to construct input files for MOSART. An attempt was made to provide all input possibilities through pulldown menus to relieve the burden of referring to lengthy tables in the User's Manual (e.g., there are 64 different materials defined in MOSART and any one of these can be selected by simply pulling down a selection menu with the mouse). Although MIB adheres to the Motif Style Guide, it is important to read this manual to gain insight on the basic operation of the tool.

8.1 System Specifications and Requirements

MIB was written in ANSI C and Motif to be as portable as possible. This version has only been compiled and tested on the Silicon Graphics Indigo, Extreme, and 4D/35. The following operating system and software libraries were used:

IRIX 4.0.5F/H and 5.2
Unix SYS V
GNU Ansi-C 2.4.5
GNU Make 3.65
Motif 1.1/1.2, X11 R4/R5

8.2 Installation

Installation of MIB is fairly simple. The first thing to do is to extract it from the delivery tape. The tape was written using 'tar' (IRIX 4.0.5H) and the command to extract all the files on tape is:

```
% tar xvo
```

or

```
% tar xvof <machine>:/dev/tape
```

This should create a directory called 'mib'. The next step is to compile and install the code:

```
% cd mib
% Install
```

This will compile the source code, install the executable 'mib' in the current directory, and will then remove all object code and associated libraries. The final step is to see if it runs:

```
% mib
```

To exit, choose 'quit' from the 'File' menu.

8.3 Problems Building MIB

The following are problems you may encounter while installing MIB:

- Code will not compile
  Are you using an Ansi-C compiler? If your Ansi-C compiler is 'gcc', edit 'Install' and change 'CC = cc' to 'CC = gcc'. Also, make sure prototypes are being used. On many systems this is 'CFLAGS = -D_PROTOTYPES'.
- Code compiles but cannot link due to unknown libraries
  Many systems have shared X11/Motif libraries (denoted -lX11_s -lXt_s -lXm_s -lPW). If yours does not, edit 'mib/src/mib/Imakefile' and change libraries to '-lXm -lXt -lX11'.

8.4 Using the MOSART Input Builder

Before using MIB, a few interface issues need to be discussed. MIB is a graphical representation of a MOSART input file and should be treated as such, i.e., the order of input groups and what you type in on the screen is what will appear in your input file when saved. As with most applications, MIB has a menu bar with three headings: File, Input Groups, and Help. These menus are accessed by positioning the mouse pointer over the menu heading and pressing (and holding) the left mouse button. When the menu appears, any item in the menu can be selected by simply positioning the pointer over the desired choice and releasing the mouse button. Bordering the window are a set of scroll bars (right and bottom) which allow areas of the form that are clipped from view to be seen by dragging the scrollbar with the left mouse button in the direction that the form is to be scrolled. There are four different styles of inputs possible in MIB: text, toggles, options, and cascade menus. Text is input by first clicking in the text box and then typing the desired text. Toggles are Yes/No, True/False buttons. Clicking on them toggles their value. The indicator on the button will illuminate to yellow when the toggle is Yes or True. Option menus look like a regular button but have a raised dash on the right side. Click with the left mouse button to pull down an option menu. A cascade button looks like a regular pushbutton, but actually has two functions. When selected with the left mouse button, it increments its value to the next item in the list. When selected with the right mouse button, a cascading menu appears to allow the user to select an option while viewing the entire list. The best way to become accustom with MIB is to follow the walk- through in Section 8.5.

Saving — Choose 'save' from the 'file' menu on the menubar. This is a standard Motif file browser. The filename defaults to the scenario name at the top of the input form. To override this name, edit the filename in the 'selection' section of the browser or click on one of the files in the file list. Directories can be traversed by double clicking on any of the directories in the directory list.

Loading Files can be loaded by choosing 'load' from the 'file' menu on the menubar. Interaction is identical to the 'save' command above. A file can also be loaded upon startup by specifying a name on the command line:

% mib test.in

Quit Choose 'quit' from the 'file' menu on the menubar to exit the program.

Groups MOSART requires four input groups to always be present: Geometry Parameters, User Specified Parameters, Spectral Parameters, and Position Parameters. Other groups can be added to the form by selecting one from the 'input group' menu on the menubar.

8.5 Walk Through

Start MIB by typing:

% mib

Text Input Fields

Single click mouse in the 'Header' text field. The cursor is positioned at that point. Insert or delete text from this point.

Double click mouse in the 'Header' text field which highlights the entire word. Typing in "Tester" replaces the word.

Triple click the mouse in the 'Header' text field. The entire field is now highlighted and anything typed replaces the entire line.

Toggle Buttons

Click on toggle button 'Terrain Temperature Calculation'. The yellow indicator will illuminate. This signifies that option is 'True' or 'Yes'. Click again the indicator turns off.

Option Buttons

Click and hold on the 'Printout Switch' option button with the left mouse button.

A popup menu will appear. Drag the mouse down to 'Medium' and release. The option button now displays 'Medium'.

Cascade Buttons

Add a new group by selecting 'Terrain Specs' from the 'Input Group' menu on the menubar. A new section will appear in the window and the system will automatically scroll down to the bottom of the form. Click on 'Terrain Index' with the left mouse button.

Notice how the label on the button changed. Now click and hold on the same button with the right mouse button. A series of pulldown menus becomes available. Clicking with the left mouse button steps through this list and clicking with the right button allows selection from the entire list.

Saving and Loading

Edit the 'Scenario Rootname' at the top of the window and select 'save' from the 'file' menu on the menubar. Notice that the scenario name is the default name on the file browser. Select 'OK' to save.

Select 'load' from the 'file' menu on the menubar. Find the file you just saved by scrolling around the file list. Once found, click once on the file (which should highlight), and select 'OK'. When the file browser goes away, the file has been loaded.

Getting Help

Select 'MOSART Manual' from the "Help" menu on the menubar. The online Mosaic version of the MOSART document will appear.

Exiting

Select 'quit' from the 'file' menu on the menubar.

APPENDIX A: INPUT TEST FILE

```
Moderate Spectral Atmospheric Radiance and Transmittance (MOSART) (Ver. 1.40)
.
User-specified Parameters -----------------------
 Header ................................ Sample Input File
 Printout Switch (S/M/L) ............... Large
 Temperature Calculations (Y/N) ........ No
 Multiple Scattering (Y/N) ............. No
 Solar/Lunar Ephemeris (Y/S/L/N) ....... No
.
Position Parameters ----------------------------
 Coordinate Reference (Obsv/Source) .... Source
 Latitude (deg) (+ North, - South) ..... 50.
 Longitude (deg) (+ East, - West) ...... -100.
 Day of the month (integer) ............ 21
 Month of the year (integer) ........... June
 Year (integer) ........................ 1991
 Time of day (24-hr HH.MMSS) ........... 12.00
 Time index (LST/GMT) .................. LST
.
Geometry Parameters ----------------------------
 Observer Azimuths (deg) (<=8) .........   0.   90.   180.   270.
 Azimuth Reference (Relative/True) ..... True
.
No. Index Obsv.Alt. Src. Alt. Sl.Rng. Earth Ang.Obsv.Angle Src. Angle Length
            (Km)       (Km)     (Km)     (deg)      (deg.)    (deg.)   Switch
 1   Se    100.0       1.0     *****     *****      *****       1.0      0
 2   Be    100.0     *****     *****     *****      *****      90.0      0
 3   Ce    100.0       1.0     *****     *****      -90.0     *****      0
 4   Ce    100.0     *****     120.0     *****       -6.0     *****      0
 5   Cz    100.0       1.0     *****       0.0      *****     *****      0
 6   Cl    100.0       1.0     *****     *****       46.0     -98.2      *
 7   Ae      1.0     *****     *****     *****      *****     *****      *
 8   He      1.0     *****       1.0     *****      *****     *****      *
 9   Le    400.0     *****     *****     *****       -7.0     *****      *
10   Le    100.0       1.0     *****     *****      *****     *****      *
11   Le    400.0     -200.     *****     *****      *****     *****      *
12   Le    400.0     *****     *****     *****      -19.75    *****      *
End of Geometry Data/
.
Spectral Parameters ----------------------------
 Spectral Calculations (MO/LO/MM) ...... MO
 Wavenumber or Wavelength (WN/WL) ...... WN
   Initial wavenumber (cm**-1/um) ...... 3000.
   Final wavenumber (cm**-1/um) ........ 3000.
   Calculation Width (cm**-1/um/GHz) ...    5.
.
```