General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

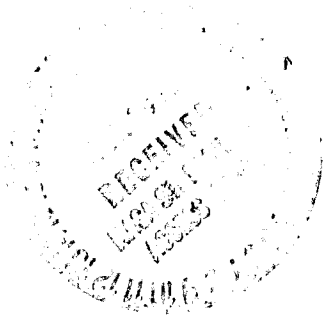
(NASA-TM-84799) SYSTEMS TECHNOLOGY LABORATORY (STL) COMPENDIUM OF UTILITIES (NASA) 110 p HC A06/MF A01 CSCL 09B

N82-29034

Unclas
G3/61 28388

SYSTEMS TECHNOLOGY LABORATORY (STL) COMPENDIUM OF UTILITIES

JULY 1981



National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt Maryland 20771

SYSTEMS TECHNOLOGY LABORATORY SERIES STL-81-003

SYSTEMS TECHNOLOGY LABORATORY (STL) COMPENDIUM OF UTILITIES

JULY 1981

NASA
National Aeronautics and
Space Administration

Goddard Space Flight Center
Greenbelt, Maryland 20771

FOREWORD

The Systems Technology Laboratory (STL) is a computational research facility located at the Goddard Space Flight Center of the National Aeronautics and Space Administration (NASA/GSFC). The STL was established in 1978 to conduct research in the area of flight dynamics systems development. The laboratory consists of a VAX-11/780 and a PDP-11/70 computer system, along with an image-processing device and some microprocessors. The operation of the Laboratory is managed by NASA/GSFC (Systems Development and Analysis Branch) and is supported by SYSTEX, Inc., Computer Sciences Corporation, and General Software Corporation.

The main goal of the STL is to investigate all aspects of systems development of flight dynamics systems (software, firmware, and hardware), with the intent of achieving system reliability while reducing total system costs. The flight dynamics systems include the following: (1) attitude determination and control, (2) orbit determination and control, (3) mission analysis, (4) software engineering, and (5) systems engineering. The activities, findings, and recommendations of the STL are recorded in the Systems Technology Laboratory Series, a continuing series of reports that includes this document. A version of this document was also issued as Computer Sciences Corporation document CSC/TM-81/6141.

The primary contributors to this document include

William J. Decker	(Computer Sciences Corporation)
Elizabeth J. Smith	(Computer Sciences Corporation)
Wayne A. Taylor	(Computer Sciences Corporation)
Phillip D. Merwarth	(Goddard Space Flight Center)
Michael E. Stark	(Goddard Space Flight Center)

Other contributors include

Rosemarna S. Pajerski	(Goddard Space Flight Center)
Frank E. McGarry	(Goddard Space Flight Center)
Arthur L. Green	(Computer Sciences Corporation)

Single copies of this document can be obtained by writing to

Keiji Tasaki
Code 582.1
NASA/GSFC
Greenbelt, Maryland 20771

ABSTRACT

This document presents a brief description of each utility contributed by users of the Systems Technology Laboratory (STL) of the Goddard Space Flight Center (GSFC) Mission Support Computing and Analysis Division (Code 580). These brief descriptions allow the reader to quickly judge the suitability of a particular utility for a particular application within the STL.

TABLE OF CONTENTS

LIST OF TABLES

Table

SECTION 1 - INTRODUCTION

This compendium provides a brief synopsis of the utilities available to users of the Goddard Space Flight Center (GSFC) Mission Support Computing and Analysis Division (Code 580) Systems Technology Laboratory (STL). The software described is a group of programs, routines, and operating system command sequences which are general in nature and can thus be used in many applications.

1.1 DEFINITION OF AN STL UTILITY

Within the STL, a utility is considered to be either a piece of software that can be included in a final product or software whose normal use in the applications environment is to assist in the operation of the final product. Thus, subroutines which perform common functions, such as data conversion or character string comparison, are considered to be utilities, since they can be included in many different types of systems. Other examples of included software are the various STL libraries of subroutines, such as graphics packages, which provide a range of functions. Utilities which assist in the operation of applications programs include programs such as those which move, convert, or reformat data files or those which simplify the user's interaction with the operating system.

Operationally, an STL utility is considered to be an extension or part of the operating system; that is, the user applies the utility in the same way as other support services supplied by the operating system. Users are encouraged to use the utilities because they are field-proven software which can be used at little cost and low risk.

In contrast to utilities, one other type of nonapplications software is recognized in the STL, the software tools (see Reference 1 for a list and descriptions of the software tools available in the STL). A software tool is considered to be

a piece of software used in the development and maintenance process to contribute quality to or reduce the cost of the software product. Rather than becoming part of the final product, a software tool is designed to act as a support service in developing the product.

Much of the software supplied by the computer manufacturer satisfies the definition of a utility given above. However, vendor-supplied utilities are not included in this compendium. Descriptions and explanations of the use of these utilities can be found in the appropriate system documentation.

The above definitions of utilities and tools are not standard and may not apply in other environments. In the STL, these definitions are used to separate the nonapplications software into manageable groups for development, maintenance, and research.

1.2 ACCESS TO STL UTILITIES

Each utility available to STL users is described in Section 2 of this document; the description for each follows the format explained at the beginning of Section 2. The presentation is intentionally brief to allow the reader to make a quick judgment as to the suitability of a particular utility to specific needs. A reader interested in a particular software utility can then find more detailed information in the documentation, source code, or sample of use cited for that entry. Users of the GSFC Code 580 STL may use or obtain the utilities from the locations cited. In almost all cases, the locations indicate files controlled by another STL user.

SECTION 2 - UTILITY DESCRIPTIONS

This section includes the individual STL utility descriptions. A general discussion of the utilities and a suggested method of using this document is given first, followed by an explanation of the formats used to present the utilities.

2.1 CLASSIFICATION OF THE STL UTILITIES

Two classification schemes for STL utilites are immediately apparent and are used in this document. First, the utilities can be classed by host computer. Currently, the STL is supported by the following two computers: (1) a Digital Equipment Corporpation (DEC) PDP-11/70 computer, operating under the RSX-11M Version 3.2 operating system and (2) a DEC VAX-11/780 computer, operating under the VMS Version 2.2 operating system. Each utility description indicates which computer supports the utility. Many of the utilities are available on both computers, and some of those utilities that are not on both computers could be moved to the other system with a minimum of effort.

A second natural classification is the form of the utility: subroutine (or function), library of multiple routines, executable program, or command procedure. The form of the utility determines the presentation format of the utility description given in this document (see Section 2.2).

A third type of classification system was developed to assist users in selecting the appropriate utilities. This scheme partitions the STL utilities into functional or applications groups. The four general categories are the following: data handling, algorithmic, graphics, and system support. These categories are described as follows:

- Data Handling--Data handling utilities are those which move data from one device to another, rearrange data from one format to another, or convert data from one

representation to another. Subroutines or libraries which provided specialized input/output (I/O) capabilities to user programs are also included in this category.

- Algorithmic--Algorithmic utilities are those which perform a standard transformation upon a set of arguments. These utilities include mathematical functions, time and data conversion routines, and functions that manipulate and compare character strings.

- Graphics--Graphics utilities provide the user with the interfaces to various display devices. The STL graphics utilities interface with a drum plotter, electrostatic printer, line printer, DEC VT100 display device, or International Imaging Systems (I^2S) Model 70CS display device.

- Systems Support--System support utilities cover a wide range of functions. Included in this category are basic debugging aids, enhancements of existing operating system utilities, utilities that provide assurance of computer-to-computer compatibility, interfaces from a program to the operating system or to the user, and other system-related functions. This category contains utilities which assist in the specific implementation of a user program.

The reader should start the search for an appropriate utility by using Tables 2-1 and 2-2; in these tables, the utilities are presented by category and subcategory for the PDP-11/70 and VAX-11/780 computers, respectively. To obtain more detailed information, the reader can then consult Table 2-3, which gives the utilities in alphabetical order by name. Once the reader has determined the utility of interest, the description of that utility can be consulted directly. These are listed alphabetically following Section 2.3.

Table 2-1. Utilities by Category on the PDP-11/70

ALGORITHMIC

Mathematical

ANDREE
URAND

Time/Date Conversion

HMS
NDAYS
YMD
YMDHMS
YRSEC

String Manipulation

MATCHS
STRING
UTCHAREQ

DATA HANDLING

Input/Output Packages

RLB11M

Data Conversion

File/Data Copy

LST
TAP
TPU
TRN

File/Data Reformat

CONVRT
EDIT
INCLUD
IOLIST
PAGER
REFORM
REMTAB
RJESOURCE
RNO
SQUEEZ
TEC
XEROX

GRAPHICS

Drum Plotter

I^2S Display Device

GRAFIN
GRAPH
IISPLT
PIE

Printer Plotter

UTGRAPH

VT100 Display Device

VT100

SYSTEM SUPPORT

Debug Aids

INDEX

Computer Compatibility

Command Language Addition

SLPPRINT
SRD

Execution Control

GETFLD
MENU
MENUC
PARA
PARAM
RQST

System Interface

RJE
STUFF

Table 2-2. Utilities by Category on the VAX-11/780

ALGORITHMIC

Mathematical

GCFIT

Time/Date Conversion

ATIME
HMS
NDAYS
YMD

String Manipulation

SEARCH

DATA HANDLING

Input/Output Packages

FTIO

Data Conversion

ASCEBC
CONVERT
DOUBLE
IODAT
SINGLE
VAXEPH

File/Data Copy

BACKUP
BLUE
TAP
TAPECOPY
TRN

File/Data Reformat

CONVRT
INDEX
REFORM
REMTAB
TEC
TIDY

GRAPHICS

Drum Plotter

CALCOMP

I^2S Display Device

Printer Plotter

PRPLOT

VT100 Display Device

VT100

SYSTEM SUPPORT

Debug Aids

BINARY
XREF

Computer Compatibility

GETADR
WAIT

Command Language Addition

CSD
DSD
KILL
LIBR
SUBMIT

Execution Control

MENU
PARA

System Interface

BATCH

Table 2-3. List of STL Utilities (1 of 4)

Utility Name	Computer(s)	Category	Subcategory	Notes
ANDREE	PDP	Algorithmic	Mathematical	Matrix inversion
ASCEBC	VAX	Data handling	Data conversion	Character translation
ATIME	VAX	Algorithmic	Time/date conversion	
BACKUP	VAX	Data handling	File/data copy	
BATCH	VAX	System support	System interface	
BINARY	VAX	System support	Debug aid	Displays integer
BLUE	VAX	Data handling	File/data copy	M&DO Xerox tapes
CALCOMP	VAX	Graphics	Drum plotter	
CONVERT	VAX	Data handling	Data conversion	IBM-to-VAX translation
CONVRT	PDP/VAX	Data handling	File/data reformat	Source module extraction
CSD	VAX	System support	Command language addition	Create directory
DOUBLE	VAX	Data handling	Data conversion	IBM-to-VAX translation
DSD	VAX	System support	Command language addition	Delete directory
EDIT	PDP	Data handling	File/data reformat	FORTRAN-callable editor
FTIO	VAX	Data handling	Input/output package	
GCFIT	VAX	Algorithmic	Mathematical	Curve fitting
GETADR	VAX	System support	Computer compatiblity	PDP-to-VAX compatibility
GETFLD	PDP	System support	Execution control	
GRAFIN	PDP	Graphics	I^2S display device	
GRAPH	PDP	Graphics	I^2S display device	

Table 2-3. List of STL Utilities (2 of 4)

Utility Name	Computer(s)	Category	Subcategory	Notes
HMS	PDP/VAX	Algorithmic	Time/date conversion	
IISPLT	PDP	Graphics	I^2S display device	
INCLUD	PDP	Data handling	File/data reformat	
INDEX	PDP	System support	Debug aid	Variable cross reference
INDEX	VAX	Data handling	File/data reformat	Compilation summary
IODAT	VAX	Data handling	Data conversion	IBM-to-VAX translation
IOLIST	PDP	Data handling	File/data reformat	
KILL	VAX	System support	Command language addition	
LIBR	VAX	System support	Command language addition	Object library creation
LST	PDP	Data handling	File/data copy	Terminal listing
MATCHS	PDP	Algorithmic	String manipulation	Character comparison
MENU	PDP/VAX	System support	Execution control	Choice selection
MENUC	PDP	System support	Execution control	Choice selection
NDAYS	PDP/VAX	Algorithmic	Time/date conversion	
PAGER	PDP	Data handling	File/data reformat	Listing formatter
PARA	PDP/VAX	System support	Execution control	Internal data editor
PARAM	PDP	System support	Execution control	Internal data editor
PIE	PDP	Graphics	I^2S display device	Pie charts

Table 2-3. List of STL Utilities (3 of 4)

Utility Name	Computer(s)	Category	Subcategory	Notes
PRPLOT	VAX	Graphics	Printer plotter	
REFORM	PDP/VAX	Data handling	File/data reformat	Listing formatter
REMTAB	PDP/VAX	Data handling	File/data reformat	
RJE	PDP	System support	System interface	PDP-11/70 to IBM S/360-95 communication
RJESOURCE	PDP	Data handling	File/data reformat	
RLBIIN	PDP	Data handling	Input/output package	
RNO	PDP	Data handling	Data/file reformat	Text formatter
RQST	PDP	System support	Execution control	
SEARCH	VAX	Algorithmic	String manipulation	
SINGLE	VAX	Data handling	Data conversion	IBM-to-VAX translation
SLPPRINT	PDP	System support	Command language addition	
SQUEEZ	PDP	Data handling	File/data reformat	
SRD	PDP	System support	Command language addition	Alphabetized directory lists
STRING	PDP	Algorithmic	String manipulation	
STUFF	PDP	System support	System interface	MCR commands from FORTRAN
SUBMIT	VAX	System support	Command language addition	
TAP	PDP/VAX	Data handling	File/data copy	
TAPECOPY	VAX	Data handling	File/data copy	
TEC	PDP/VAX	Data handling	File/data reformat	Character editor

Table 2-3. List of STL Utilities (4 of 4)

Utility Name	Computer(s)	Category	Subcategory	Notes
TIDY	VAX	Data handling	File/data reformat	FORTRAN source code formatter
TPU	PDP	Data handling	File/data copy	
TRN	PDP/VAX	Data handling	File/data copy	
URAND	PDP	Algorithmic	Mathematical	Random numbers
UTCHAREQ	PDP	Algorithmic	String manipulation	
UTGRAPH	PDP	Graphics	Printer plotter	
VAXEPH	VAX	Data handling	Data conversion	
VT100	PDP/VAX	Graphics	VT100 display device	
WAIT	VAX	System support	Computer compatibility	PDP-to-VAX compatibility
XEROX	PDP	Data handling	File/data reformat	
XREF	VAX	System support	Debug aid	FORTRAN cross reference
YMD	PDP/VAX	Algorithmic	Time/date conversion	
YMDHMS	PDP	Algorithmic	Time/date conversion	
YRSEC	PDP	Algorithmic	Time/date conversion	

2.2 STL UTILITY DESCRIPTION FORMATS

The description of each STL utility given in this document starts with a brief description of the function of the utility, followed by its file locations, language, software type, and list of available documentation.

As noted in Section 2.1, the utilities can be classified by form or type (subroutine, library of multiple routines, executable program, or command procedure). The appropriate designation is given under TYPE in the description of each utility. The type of utility determines whether certain items, such as calling sequences or help files, appear in the description.

The first part of each description is a page header indicating the utility name, host computer, and date at which the information was collected. The DESCRIPTION of the utility appears directly under the page header and is a brief statement about the intended use of the utility. The description of a library usually includes some mention of a representative sample or range of the functions performed by the individual modules within the library. The description of a subroutine or function also includes the calling sequence and a description of the calling sequence arguments.

The files required to use and understand the utility are listed under LOCATION. Each file location is given in a fully qualified format which includes the device name. For subroutines, the location is given for the source code. Task image locations are indicated for programs, except for those programs which have been "installed" on either the PDP-11/70 or the VAX-11/780. The location of the object library (OLB file) is given for libraries. For command procedures, the COM or CMD file locations are given.

For some utilities, a file showing the use of the utility in a representative context is given under the LOCATION heading. For subroutines, this Sample of Use might be a test program which calls the subroutine. For programs, it might be a command procedure which indicates the proper setup before the program is run. Similarly, an informational Help File may be given for a program or command procedure utility, since these are used interactively.

The LANGUAGE(S) in which the utility is written is given next, to indicate to the reader the amount of effort which might be required to move the utility to another STL computer.

Finally, the available DOCUMENTATION for each utility is listed. The level of documentation is not uniform: some utilities are well documented with user's guides and system descriptions, others are documented in source code prologs, and many are not documented outside of the description given in this compendium.

2.3 STL UTILITY DESCRIPTIONS

The following pages in this section present the individual STL utility descriptons. The utilities are given in alphabetical order by name. Utilities which are available on both the PDP-11/70 and the VAX-11/780 are described separately for each computer.

DESCRIPTIONS OF THE STL UTILITIES

UTILITY NAME (COMPUTER): ANDREE (PDP)

DESCRIPTION:

ANDREE computes the rank and pseudoinverse of a symmetric matrix using a modified ANDREE algorithm.

CALLING SEQUENCE: CALL ANDREE(V,N,NR,EPS,B,U,R,MM,ISV)

Parameter	Type	Dimension	I/O	Description
V	R*8	MM	I/O	Matrix and its inverse
N	I*2	1	I	Number of rows (columns) in matrix V
NR	I*2	1	O	Rank of matrix V
EPS	R*8	1	I	Number of accurate digits
B	R*8	MM	-	Local storage array
U	R*8	MM,MM	-	Local storage array
R	R*8	MM	-	Local storage array
MM	I*2	1	I	Dimension for the B, V, U, and R arrays
ISV	I*2	1	I	Logical unit number for temporary storage area

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[210,1]ANDREE.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Routine prolog

Least-Squares and Pseudoinversion, GSFC Doc. No. X-551-69-73, E. Lefferts, September 1, 1968

UTILITY NAME (COMPUTER): ASCEBC (VAX)

DESCRIPTION:

ASCEBC converts ASCII characters to EBCDIC.

CALLING SEQUENCE: CALL ASCEBC(STRING1,STRING2,LEN)

Parameter	Type	Dimension	I/O	Description
STRING1	B	LEN	I	ASCII character input
STRING2	B	LEN	O	EBCDIC character output
LEN	I*4	1	I	Length of input strings in bytes

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[FDYN.FDY03]ASCEBC.FOR

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): ATIME (VAX)

DESCRIPTION:

ATIME converts numeric time in any units to the form "DD HH:MM:SS:CC" (VAX time).

CALLING SEQUENCE: CALL ATIME(ITIME,ITYPE,CTIME,IRET)

Parameter	Type	Dimension	I/O	Description
ITIME	I*4	1	I	Input time
ITYPE	I*4	1	I	Units-of-time flag: = 0, system ticks (nanoseconds) = 1, milliseconds = 2, seconds = 3, minutes = 4, hours
CTIME	C*14	1	O	ASCII string containing time
IRET	I*4	1	O	Return code: = 0, successful = 1, input time greater than 1000 days

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[FDYN.FDY05.WAIT]ATIME.FOR

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): BACKUP (VAX)

DESCRIPTION:

BACKUP backs up all files (including main directories and subdirectories) for any VAX user.

LOCATION:

	Disk:[Directory]Name
COM File:	DBB1:[FDYN.FDY05]BACKUP.COM
Sample of Use:	None
Help File:	None

LANGUAGE: Digital Command Language (DCL)

TYPE: Command procedure

DOCUMENTATION: DBB1:[FDYN.FDY05]BACKUP.COM

UTILITY NAME (COMPUTER): BATCH (VAX)

DESCRIPTION:

BATCH determines whether the process being executed is interactive or batch.

CALLING SEQUENCE: CALL BATCH(LFLAG)

Parameter	Type	Dimension	I/O	Description
LFLAG	L*4	1	O	Job mode determination flag: = .TRUE. , batch job = .FALSE., interactive job

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[FDYN.FDY05.CONVERT]BATCH.FOR

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): BINARY (VAX)

DESCRIPTION:

BINARY produces a binary representation of an integer at the terminal for debugging purposes.

CALLING SEQUENCE: CALL BINARY(INT)

Parameter	Type	Dimension	I/O	Description
INT	I*4	1	I	Integer to be displayed

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[FDYN.FDY03.STUFF]BINARY.SRV

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): BLUE (VAX)

DESCRIPTION:

BLUE generates FORTRAN and assembler listings of all FORTRAN/assembler files in a directory and generates a tape to be read on the IBM S/360 computer to produce "blue-book" (M&DO Xerox) listings.

LOCATION:

	Disk:[Directory]Name
COM File:	DBB1:[FDYN.FDY05.BLUE]BLUE.COM
Sample of Use:	None
Help File:	None

LANGUAGE: Digital Command Language (DCL)

TYPE: Command procedure

DOCUMENTATION: None

UTILITY NAME (COMPUTER): CALCOMP (VAX)

DESCRIPTION:

CALCOMP generates CalComp plots. This package contains all of the standard routines available from CalComp.

LOCATION:

	Disk:[Directory]Name
Object Library:	DBB1:[FDYN]PLOT.OLB
Sample of Use:	None

LANGUAGE(S): FORTRAN

TYPE: Library

DOCUMENTATION: "Programming CalComp Electromechanical Plotters," California Computing Products, Inc., October 1975

UTILITY NAME (COMPUTER): CONVERT (VAX)

DESCRIPTION:

CONVERT changes unformatted IBM-S/360 data to internal VAX representation. The type of data is user specified. A call to an ENTRY point (SET_UP) is required prior to the call to CONVERT.

CALLING SEQUENCE: CALL SET_UP(INBUF,OUTBUF)
CALL CONVERT(INP)

Parameter	Type	Dimension	I/O	Description
INBUF	Any	*	I	Buffer containing the IBM S/360 formatted data
OUTBUF	Any	*	O	Buffer to receive the VAX-11 formatted data
INP	I*4	1	I	Index indicating the type of data to be converted

*The calling routine should dimension both of these arrays to match the record size of the data being converted.

LOCATION:

Disk:[Directory]Name

Source Code: DBA0:[GMAS]CONVERT.MAR

Sample of Use: None

LANGUAGE: MACRO

TYPE: Subroutine

DOCUMENTATION: "Guide for Converting Software From System 360 to VAX-11/780," memorandum from J. S. Watson, GSFC Code 580, April 1981

UTILITY NAME (COMPUTER): CONVRT (PDP)

DESCRIPTION:

CONVRT decomposes large sequential FORTRAN source files into individual subroutines.

LOCATION:

	Disk:[Directory]Name
Image:	DB0:[203,5]CONVRT.TSK
Sample of Use:	None
Help File:	(see CONVRT(VAX) description)

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: (see CONVRT(VAX) description)

UTILITY NAME (COMPUTER): CONVRT (VAX)

DESCRIPTION:

CONVRT decomposes large sequential FORTRAN source files into individual subroutines.

LOCATION:

	Disk:[Directory]Name
Image:	DBB1:[FDYN.FDY05.CONVRT]CONVRT.EXE
Sample of Use:	None
Help File:	DBB1:[FDYN.FDY05.CONVRT]INSTRUCT.DAT

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: DBB1:[FDYN.FDY05.CONVRT]INSTRUCT.DAT

UTILITY NAME (COMPUTER): CSD (VAX)

DESCRIPTION:

CSD creates subdirectories.

LOCATION:

	Disk:[Directory]Name
.COM File:	DBB1:[FDYN.FDY05]CREATESD.COM
Sample of Use:	None
Help File:	None

LANGUAGE: Digital Command Language (DCL)

TYPE: Command procedure

DOCUMENTATION: None

UTILITY NAME (COMPUTER): DOUBLE (VAX)

DESCRIPTION:

DOUBLE converts unformatted IBM S/360 double-precision data to internal VAX-11/780 representation.

CALLING SEQUENCE: DOUT = DOUBLE(DINP)

Parameter	Type	Dimension	I/O	Description
DINP	R*8	1	I	REAL*8 value in IBM S/360 internal format. Before passing this value to DOUBLE, the calling program must reverse the order of bytes in both long words
DOUT	R*8	1	O	REAL*8 value in VAX-11 internal format

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[GMAS]DOUBLE.MAR

Sample of Use: None

LANGUAGE: MACRO

TYPE: Subroutine

DOCUMENTATION: "Guide for Converting Software From System 360 to VAX-11/780," memorandum from J. S. Watson, GSFC Code 580, April 1981

UTILITY NAME (COMPUTER): DSD (VAX)

DESCRIPTION:

DSD deletes subdirectories in a consistent manner.

LOCATION:

	Disk:[Directory]Name
COM File:	DBB1:[FDYN.FDY05]DELETESD.COM
Sample of Use:	None
Help File:	None

LANGUAGE: Digital Command Language (DCL)

TYPE: Command procedure

DOCUMENTATION: None

UTILITY NAME (COMPUTER): EDIT (PDP)

DESCRIPTION:

EDIT is a set of FORTRAN routines which performs all of the common functions of the PDP text editor. The command syntax is the same as that of the PDP text editor. This set of routines enables a user program to control a text editor.

LOCATION:

Disk:[Directory]Name

Object Library: Because the edit routines should be overlayed, there is no object library. The following routines can be found in UIC DB0:[206,2]:

ADD.FTN	INS.FTN	REMVB.FTN
ADDP.FTN	KILL.FTN	RET.FTN
BOT.FTN	LOC.FTN	SAV.FTN
CHG.FTN	LSTP.FTN	SCHG.FTN
COMPL.FTN	LSTT.FTN	TLOC.FTN
DEL.FTN	MOVE.FTN	TOP.FTN
DELP.FTN	NEX.FTN	TYP.FTN
EDIT.FTN	NEXP.FTN	UNS.FTN
EXT.FTN	OVE.FTN	WRT.FTN
GETLIN.FTN	PRI.FTN	

Sample of Use: None

LANGUAGE(S): FORTRAN

TYPE: Library

DOCUMENTATION: Subroutine descriptions of the EDIT routines can be found in the following document:

Configuration Analysis Tool (CAT) System Description and User's Guide (CSC/SD-80/6089)

UTILITY NAME (COMPUTER): FTIO (VAX)

DESCRIPTION:

FTIO is a group of routines which can be used to perform I/O on foreign tapes on the VAX-11/780. The routines are most commonly used to read tapes from the IBM S/360 computers or to write tapes to be taken to the IBM computers.

- LOCATION:

	Disk:[Directory]Name
Source Code:	DBB1:[GMAS]FTIO.MAR
Sample of Use:	None

LANGUAGE(S): MACRO

TYPE: Library

DOCUMENTATION: "Guide for Converting Software From System 360 to VAX-11/780," memorandum from J. S. Watson, GSFC Code 580, April 1981

UTILITY NAME (COMPUTER): GCFIT (VAX)

DESCRIPTION:

GCFIT is a generalized curve-fitting program. It fits a curve through given X-Y data. Options are provided to permit CalComp plots to be produced.

LOCATION:

	Disk:[Directory]Name
Image:	DBB1:[FDYN.FDY05.GCFIT]GCFIT.EXE
Sample of Use:	None

LANGUAGE: FORTRAN

TYPE: Program

DOCUMENTATION: Generalized Curve Fitting Program (GCFIT) Analytic Description and User's Guide, January 1970

UTILITY NAME (COMPUTER): GETADR (VAX)

DESCRIPTION:

GETADR fills an input array with the addresses of arguments in a calling sequence. GETADR performs the same function as the system directive GETADR on the PDP-11/70.

CALLING SEQUENCE: CALL GETADR(IPRM,ARG1,...,ARGN)

Parameter	Type	Dimension	I/O	Description
IPRM	I*4	N	O	Array to receive the addresses of the following arguments
ARG1	Any	1	I	First argument whose address will go into IPRM
⋮	⋮	⋮	⋮	⋮
ARGN	Any	1	I	Last (Nth) argument whose address will go into IPRM

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[FDYN.FDY03]GETADR.MAR

Sample of Use: None

LANGUAGE: MACRO

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): GETFLD (PDP)

DESCRIPTION:

GETFLD displays the given prompt and returns the user-entered character string. If an indirect file is entered (@FILE), it is then read. This utility allows user programs to read indirect files when @FILE is entered in response to a prompt.

CALLING SEQUENCE: CALL GETFLD(TEXT,EXTFIL,FLDLEN,TERMNL, EOFTTY,ERROR,FIELD)

Parameter	Type	Dimension	I/O	Description
TEXT	L*1	*	I	Prompt
EXTFIL	I*2	1	I	Unit number of the external file to be read
FLDLEN	I*2	1	I	Length of the field
TERMNL	L*1	1	I/O	Flag which indicates whether the terminal or external file is to be read
EOFTTY	L*1	1	I/O	End of file (EOF) on the terminal flag
ERROR	L*1	1	I/O	Error flag
FIELD	L*1	FLDLEN	O	Field to be obtained

*This argument is a character string terminated by a '#' character.

LOCATION:

Disk:[Directory]Name

Source Code: DB1:[204,7]UTGETFLD.FPP
DB1:[204,7]UTWHERE.FPP

Sample of Use: None

LANGUAGE: Structured FORTRAN

TYPE: Subroutine

DOCUMENTATION: Routine prolog

UTILITY NAME (COMPUTER): GRAFIN (PDP)

DESCRIPTION:

GRAFIN is a set of routines that simplifies a call to the graphics routines (e.g., POINT, VECTOR). This utility interfaces with the I^2S Generalized Graphics Utility, GRAPH.

LOCATION:

	Disk:[Directory]Name
Source Code:	DB1:[204,12]GRname.FPP
Sample of Use:	None

LANGUAGE(S): Structured FORTRAN

TYPE: Library

DOCUMENTATION: None

UTILITY NAME (COMPUTER): GRAPH (PDP)

DESCRIPTION:

GRAPH is the I^2S Generalized Graphics Utility. It provides plots and character generation on the I^2S graphics device.

LOCATION:

	Disk:[Directory]Name
Object Library:	DB0:[203,2]GRAPH.OLB
Sample of Use:	None

LANGUAGE(S): FORTRAN

TYPE: Library

DOCUMENTATION: International Imaging Systems (I^2S) Generalized Graphic Utilities User's Guide (CSC/SD-76/6078)

UTILITY NAME (COMPUTER): HMS (PDP)

DESCRIPTION:

HMS converts a packed time in hours, minutes, and seconds (HHMMSS.SS) to three variables containing hours and minutes and seconds from 0 hours of day.

CALLING SEQUENCE: CALL HMS(T,IH,IM,S)

Parameter	Type	Dimension	I/O	Description
T	R*4	1	I	Input time in packed format HHMMSS.SS
IH	I*4	1	O	Hour from the HH field of HHMMSS.SS
IM	I*4	1	O	Minute from the MM field of HHMMSS.SS
IY	R*4	1	O	Seconds from 0 hours of the day

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[207,1]HMS.FTN .

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Onboard Navigation Package (ONPAC) Simulator User's Guide (CSC/SD-81/6029)

UTILITY NAME (COMPUTER): HMS (VAX)

DESCRIPTION:

HMS converts a packed time in hours, minutes, and seconds (HHMMSS.SS) to three variables containing hours and minutes and seconds from 0 hours of day.

CALLING SEQUENCE: CALL HMS(T,IH,IM,S)

Parameter	Type	Dimension	I/O	Description
T	R*4	1	I	Input time in packed format HHMMSS.SS
IH	I*4	1	O	Hour from the HH field of HHMMSS.SS
IM	I*4	1	O	Minute from the MM field of HHMMSS.SS
IY	R*4	1	O	Seconds from 0 hours of the day

LOCATION:

Disk:[Directory]Name

Source Code: DBA0:[LNDSAT.ONPAC]HMS.FOR

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Onboard Navigation Package (ONPAC) Simulator User's Guide (CSC/SD-81/6029)

UTILITY NAME (COMPUTER): IISPLT (PDP)

DESCRIPTION:

A call to IISPLT copies the current image from the I^2S graphics device to the printer/plotter.

CALLING SEQUENCE: CALL IISPLT

LOCATION:

	Disk:[Directory]Name
Source Code:	DB0:[207,1]IISPLT.FTN
Sample of Use:	None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): INCLUD (PDP)

DESCRIPTION:

INCLUD expands INCLUDE statements in FORTRAN source code and generates a source listing. INCLUD also recognizes and eliminates tab characters.

LOCATION:

	Disk:[Directory]Name
Image:	DB0:[207,1]INCLUD.TSK
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): INDEX (PDP)

DESCRIPTION:

INDEX produces a cross-reference listing of variables in a FORTRAN source code module.

LOCATION:

	Disk:[Directory]Name
Image:	DB1:[202,2]INDEX.TSK
Sample of Use:	None
Help File:	DB1:[202,2]INDEX.HLP

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: DB1:[202,2]INDEX.DOC

UTILITY NAME (COMPUTER): INDEX (VAX)

DESCRIPTION:

INDEX reads a FORTRAN compiler output listing and creates an index listing file containing module names and their starting page number within the compiler listing. INDEX is particularly useful when the number of modules is large.

LOCATION:

	Disk:[Directory]Name
Image:	DBB1:[ADGEN.UTILITY]INDEX.EXE
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): IODAT (VAX)

DESCRIPTION:

IODAT is a BLOCK DATA module containing a translation table from EBCDIC to ASCII. The table is in a format which can be used with the MACRO MOVTC instruction or with the FORTRAN common run time routine, LIB$MOVTC.

CALLING SEQUENCE: COMMON/CHARTBL/ZEROS(256),TABLE(256)

Parameter	Type	Dimension	I/O	Description
ZEROS	Byte	256	-	Zero value bytes
TABLE	Byte	256	-	Character translation table

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[GMAS]IODAT.MAR

Sample of Use: None

LANGUAGE: MACRO

TYPE: Subroutine (BLOCK DATA)

DOCUMENTATION: "Guide for Converting Software From System 360 to VAX-11/780," memorandum from J. S. Watson, GSFC Code 580, April 1981

UTILITY NAME (COMPUTER): IOLIST (PDP)

DESCRIPTION:

IOLIST reads a FORTRAN source routine, copies the calling sequence to the prolog, and includes variable descriptions from an external file.

LOCATION:

	Disk:[Directory]Name
Image:	DB1:[204,12]IO.TSK
Sample of Use:	None
Help File:	None

LANGUAGE(S): Structured FORTRAN

TYPE: Program

DOCUMENTATION: Prolog of IOLIST MAIN routine (location = DB1:[204,12]IOLIST.FPP)

UTILITY NAME (COMPUTER): KILL (VAX)

DESCRIPTION:

KILL saves the printout LOG file from a batch job on disk, rather than sending it to the line printer at the completion of the job. The LOG file will be found under the user's LOGIN default directory.

KILL must be the last statement in a submitted command file.

LOCATION:

	Disk:[Directory]Name
COM File:	DBB1:[FDYN.FDY05.KILL]KILL.COM
Sample of Use:	None
Help File:	None

LANGUAGE: Digital Command Language (DCL)

TYPE: Command procedure

DOCUMENTATION: None

UTILITY NAME (COMPUTER): LIBR (VAX)

DESCRIPTION:

LIBR creates/updates an object library from the objects on the default directory.

LOCATION:

	Disk:[Directory]Name
COM File:	DBB1:[FDYN.FDY05]LIB.COM
Sample of Use:	None
Help File:	None

LANGUAGE: Digital Command Language (DCL)

TYPE: Command Procedure

DOCUMENTATION: DBB1:[FDYN.FDY05]LIB.COM

UTILITY NAME (COMPUTER): MATCHS (PDP)

DESCRIPTION:

MATCHS is a FORTRAN-callable function that compares two strings.

CALLING SEQUENCE: SAME = MATCHS(ARRAY1,ARRAY2,NBYTES)

Parameter	Type	Dimension	I/O	Description
ARRAY1	L*1	NBYTES	I	First string to be compared
ARRAY2	L*1	NBYTES	I	Second string to be compared
NBYTES	I*2	1	I	Number of bytes in the strings to be compared
SAME	L*4	1	O	String comparison flag: = .TRUE. , string 1 = string 2 = .FALSE., string 1 ≠ string 2

LOCATION:

Disk:[Directory]Name

Source Code: DB1:[204,7]UMATCHS.FPP

Sample of Use: None

LANGUAGE: Structured FORTRAN

TYPE: Subroutine

DOCUMENTATION: Software Engineering Laboratory (SEL) Data Base Organization and User's Guide (CSC/SD-81/6011)

UTILITY NAME (COMPUTER): MENU (PDP)

DESCRIPTION:

MENU displays a menu at the terminal containing 1 to 20 items (in two columns, if there are more than 10 items). MENU also includes a display time-out feature.

CALLING SEQUENCE: CALL MENU(LUN,IRCODE,HDR,ENT1,...,ENTN)

Parameter	Type	Dimension	I/O	Description
LUN	I*2	1	I	Logical unit number for the terminal
IRCODE	I*2	1	O	Return code: >0, menu selection <0, error
HDR	Any	*	I	Page header (maximum of 70 characters)
ENT1	Any	*	I	Menu entry 1
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
ENTN	Any	*	I	Menu entry N

*These arguments are character strings terminated by a null byte.

LOCATION:

	Disk:[Directory]Name
Source Code:	DB0:[206,2]MENU.MAC
Sample of Use:	None

LANGUAGE: MACRO

TYPE: Subroutine

DOCUMENTATION: AODS/ADEPT User's Guide (CSC/SD-81/6081)

UTILITY NAME (COMPUTER): MENU (VAX)

DESCRIPTION:

The MENU library contains routines that can be used to create displays from which selections are made. One routine within the library (MENU) imitates an older version of the PDP MENU routine. Other routines can be used to customize the selection display through the use of nonselectable explanatory text.

LOCATION:

	Disk:[Directory]Name
Object Library:	DBB1:[FDYN.FDY03]MENGEN.OLB
Sample of Use:	DBB1:[FDYN.FDY03]MENDEMO.WJD

LANGUAGE(S): FORTRAN, MACRO

TYPE: Library

DOCUMENTATION: "VAX Version of MENU Routine", memorandum to the CSC Task 990 file, September 15, 1980

UTILITY NAME (COMPUTER): MENUC (PDP)

MENUC displays a menu of up to 20 alternatives. The user's choice may be used to alter the execution path of a program. MENUC is written entirely in FORTRAN, thus providing computer transportability.

CALLING SEQUENCE: ICHOYS = MENUC(UNIT,NITEM,TITLE,PROMPT, NSTART,ITM1,...,ITMN)

Parameter	Type	Dimension	I/O	Description
UNIT	I*2	1	I	Logical unit number assigned to the terminal
NITEM	I*2	1	I	Number of items to be displayed
TITLE	Any	*	I	Character string for the display title
PROMPT	Any	*	I	Character string for the user prompt
NSTART	I*2	1	I	Sequence number of ITM1
ITM1	Any	*	I	Character string describing the first alternative
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
ITMN	Any	*	I	Character string describing the NITEMth alternative
ICHOYS	I*2	1	O	Sequence number of the selected item (NSTART-1 is returned if no selection is made)

*All character strings are variable in length and must be terminated with the character '@' (which is not displayed).

LOCATION:

	Disk:[Directory]Name
Object Code:	DB1:[216,2]SYSLIB.OLB
Sample of Use:	None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): NDAYS (PDP)

DESCRIPTION:

NDAYS converts a packed date in year, month, and day (YYMMDD.) to time in days since 1900.

CALLING SEQUENCE: CALL NDAYS(TIMYMD,INDAYS)

Parameter	Type	Dimension	I/O	Description
TIMYMD	R*8	1	I	Packed date in YYMMDD.
INDAYS	I*4	1	O	Date in TIMYMD expressed in days from 1900

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[207,1]NDAYS.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Onboard Navigation Package (ONPAC) Simulator User's Guide (CSC/SD-81/6029)

UTILITY NAME (COMPUTER): NDAYS (VAX)

DESCRIPTION:

NDAYS converts a packed date in year, month, and day (YYMMDD.) to time in days since 1900.

CALLING SEQUENCE: CALL NDAYS(TIMYMD,INDAYS)

Parameter	Type	Dimension	I/O	Description
TIMYMD	R*8	1	I	Packed date in YYMMDD.
INDAYS	I*4	1	O	Date in TIMYMD expressed in days from 1900

LOCATION:

Disk:[Directory]Name

Source Code: DBA0:[LNDSAT.ONPAC]NDAYS.FOR

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Onboard Navigation Package (ONPAC) Simulator User's Guide (CSC/SD-81/6029)

UTILITY NAME (COMPUTER): PAGER (PDP)

DESCRIPTION:

PAGER paginates listing (documentation) files for report-style output. It can also convert printing to lower case.

LOCATION:

	Disk:[Directory]Name
Image:	DB1:[204,6]PAGER.TSK
Sample of Use:	DB1:[204,5]PAGER.HLP

LANGUAGE(S): Pascal

TYPE: Program

DOCUMENTATION: Prolog of PAGER MAIN routine (location = DB1:[204,6]PAGER.PAS)

UTILITY NAME (COMPUTER): PARA (PDP)

DESCRIPTION:

PARA displays and edits parameters. Each parameter is displayed on a separate line with a parameter number, description, and value. The parameter can be protected or unprotected.

PARA differs from the PARAM utility in that it is most easily used when the parameters are not in consecutive core locations.

CALLING SEQUENCE: CALL PARA(LUN,IRCODE,HDR,NPARM,NCHAR, IDESC,IADDR,ITYPE,IFMT)

Parameter	Type	Dimension	I/O	Description
LUN	I*2	1	I	Logical unit number of the terminal
IRCODE	I*2	1	O	Return code: = 0, normal return (no editing) = 1, normal return (with editing) = -1, invalid parameter = -2, error
HDR	L*1	*	I	Variable-length header for the display
NPARM	I*2	1	I	Number of parameters to be displayed
NCHAR	I*2	1	I	Length of descriptor (in characters)
IDESC	L*1	NCHAR, NPARM	I	Array of parameter descriptors
IADDR	I*2	NPARM	I	Array of addresses for locating parameters
ITYPE	I*2	NPARM	I	Array of parameter types
IFMT	I*2	3, NPARM	I	Array of format specifications for parameters (no parentheses)

*This argument is a character string terminated by a null byte.

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[206,2]PARA.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Navigation Package (NAVPAK) System Description (CSC/SD-76/6094)

Manpower Allocation and Reporting System (MARS) System Description and User's Guide (CSC/TM-77/6295) (user's information given on pages 3-1 through 3-3)

UTILITY NAME (COMPUTER): PARA (VAX)

DESCRIPTION:

PARA displays and edits parameters. Each parameter is displayed on a separate line with a parameter number, description, and value. The parameter can be protected or unprotected.

The VAX version of PARA differs from the PDP version, as is explained in the documentation.

CALLING SEQUENCE: CALL PARA(LUN,IRCODE,HDR,NPARM,NCHAR, IDESC,IADDR,ITYPE,IFMT)

Parameter	Type	Dimension	I/O	Description
LUN	I*2	1	I	Logical unit number of the terminal
IRCODE	I*2	1	O	Return code: = 0, normal return (no editing) = 1, normal return (with editing) = -1, invalid parameter = -2, error
HDR	L*1	*	I	Variable-length header for the display
NPARM	I*2	1	I	Number of parameters to be displayed
NCHAR	I*2	1	I	Length of descriptor (in characters)
IDESC	L*1	NCHAR, NPARM	I	Array of parameter descriptors
IADDR	I*2	NPARM	I	Array of addresses for locating parameters
ITYPE	I*2	NPARM	I	Array of parameter types
IFMT	I*2	3, NPARM		

*This argument is a character string terminated by a null byte.

LOCATION:

	Disk:[Directory]Name
Object Code:	DBB1:[FDYN.FDY03]MENGEN.OLB
Sample of Use:	DBB1:[FDYN.FDY03]PARTST.WJD

LANGUAGE: FORTRAN, MACRO

TYPE: Subroutine

DOCUMENTATION: "Vax Version of PARA Routine," memorandum to CSC Task 990 File, September 15, 1980

UTILITY NAME (COMPUTER): PARAM (PDP)

DESCRIPTION:

PARAM displays and edits parameters. Each parameter is displayed on a separate line with a parameter number, description, and value. The parameter can be protected or unprotected.

PARAM differs from the PARA utility in that it is most easily used when all parameters are in consecutive core locations.

CALLING SEQUENCE: CALL PARAM(LUN,IRCODE,HDR,NPARM,IADDR, IDESC,IOFF,ITYPE,IFMT)

Parameter	Type	Dimension	I/O	Description
LUN	I*2	1	I	Logical unit number of the terminal
IRCODE	I*2	1	O	Return code: = 0, normal return (no editing) = 1, normal return (with editing) = -1, invalid parameter = -2, error
HDR	L*1	*	I	Variable-length header for the display
NPARM	I*2	1	I	Number of parameters to be displayed
IADDR	I*2	1	I	Address of the parameter record
IDESC	I*2	NPARM	I	Array of 12-character parameter descriptors
IOFF	I*2	NPARM	I	Array of offsets for locating parameters
ITYPE	I*2	NPARM	I	Array of parameter types
IFMT	I*2	3, NPARM	I	Array of format specifications (no parentheses)

*This argument is a character string terminated by a null byte.

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[206,2]PARAM.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Navigation Package (NAVPAK) System Description (CSC/SD-76/6094)

Manpower Allocation and Reporting System (MARS) System Description and User's Guide (CSC/TM-77/6295) (user's information is given on pages 3-1 through 3-3)

UTILITY NAME (COMPUTER): PIE (PDP)

DESCRIPTION:

PIE reads a data file and creates a pie chart of the data on the I^2S graphics device.

LOCATION:

	Disk:[Directory]Name
Image:	DB1:[204,12]PI.TSK
Sample of Use:	None
Help File:	None

LANGUAGE(S): Structured FORTRAN

TYPE: Program

DOCUMENTATION: Prolog of the PIE MAIN routine (location = DB1:[204,12]PIDRIV.FPP)

UTILITY NAME (COMPUTER): PRPLOT (VAX)

DESCRIPTION:

PRPLOT generates printer plots of numeric information. The plot size can be adjusted so that it fits on a CRT screen or fills an entire printed page.

LOCATION:

	Disk:[Directory]Name
Object Library:	DBB1:[FDYN.FDY05.PRPLOT]PRPLOT.OBJ
Sample of Use:	None

LANGUAGE(S): FORTRAN

TYPE: Library

DOCUMENTATION: M&DO O/S 360 User's Guide, Vol. II
(GSFC publication, January 30, 1978)

UTILITY NAME (COMPUTER): REFORM (PDP)

DESCRIPTION:

REFORM prints ASCII files on the line printer, 50 lines per page. Line numbers are printed along the extreme right margin. Page headers contain the name of the file, date, and page number.

LOCATION:

	Disk:[Directory]Name
Image:	DB1:[216,1]REFORM.TSK
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): REFORM (VAX)

DESCRIPTION:

REFORM prints ASCII files on the line printer, 50 lines per page. Line numbers are printed along the extreme right margin. Page headers contain the name of the file, date, and page number.

LOCATION:

	Disk:[Directory]Name
Image:	DBB1:[FDYN.FDY02]REFORM.EXE
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): REMTAB (PDP)

DESCRIPTION:

REMTAB is a utility program that removes tabs and control characters from an input text file. An output file of the same name but with an incremented version number or a new file with a different name is created at the request of the user.

LOCATION:

	Disk:[Directory]Name
Image:	DB0:[201,2]REMTAB.TSK
Sample of Use:	None
Help File:	(see REMTAB(VAX) description)

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: (see REMTAB(VAX) description)

UTILITY NAME (COMPUTER): REMTAB (VAX)

DESCRIPTION:

REMTAB is a utility program that removes tabs and control characters from an input text file. An output file of the same name but with an incremented version number or a new file with a different name is created at the request of the user.

LOCATION:

	Disk:[Directory]Name
Image:	DBA0:[DEBATT.DECAP]REMTAB.EXE
Sample of Use:	None
Help File:	DBB1:[GRES.BOB]REMTAB.DOC

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: Disk file DBB1:[GRES.BOB]REMTAB.DOC

UTILITY NAME (COMPUTER): RJE (PDP)

DESCRIPTION:

RJE submits batch jobs to the M&DO IBM S/360-95 computer and returns the job output when it becomes available.

LOCATION:

	Disk:[Directory]Name
Image:	Installed as RJE command
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN, MACRO

TYPE: Program

DOCUMENTATION: "PDP-11/70 RJE User's Guide," memorandum from GSFC Code 580 (undated)

UTILITY NAME (COMPUTER): RJESOURCE (PDP)

DESCRIPTION:

RJESOURCE processes RJE output to eliminate carriage controls, tabs, and numbers in columns 73-80.

LOCATION:

	Disk:[Directory]Name
Image:	DB0:[201,2]RJESOURCE.TSK
Sample of Use:	None
Help File:	DB0:[201,2]RJESOURCE.DOC

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: DB0:[201,2]RJESOURCE.DOC

UTILITY NAME (COMPUTER): RLB11M (PDP)

DESCRIPTION:

RLB11M reads logical blocks from tape. RLB11M will read logical blocks which are not acceptable to the TRN utility because of their size.

CALLING SEQUENCE: CALL RLB11M(LUN,INBUF,BUFLEN,EOF,ERR)

Parameter	Type	Dimension	I/O	Description
LUN	I*2	1	I	Logical unit number of the tape
INBUF	L*1	BUFLEN	I	Input buffer
BUFLEN	I*2	1	I	Buffer length
EOF	L*2	1	O	End of file (EOF) flag
ERR	L*2	1	O	Error flag

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[201,2]RLB11M.FTN

Sample of Use: DB0:[201,2]TSTRLB11M.FTN

LANGUAGE: Structured FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): RNO (PDP)

DESCRIPTION:

RNO formats text. Operations provided include justifying margins, paragraphing, line counting, etc.

LOCATION:

	Disk:[Directory]Name
Image:	DB0:[1,54]RNO.TSK
Sample of Use:	None
Help File:	None

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: DB0:[201,3]RNO.DOC

UTILITY NAME (COMPUTER): RQST (PDP)

DESCRIPTION:

The RQST library consists of five interactive routines which prompt (request) the user for one of the following: date, integer, yes/no, or two kinds of strings.

LOCATION:

	Disk:[Directory]Name
Object Library:	DB1:[216,2]STRLIB.OLB
Sample of Use:	None

LANGUAGE(S): FORTRAN

TYPE: Library

DOCUMENTATION: None

UTILITY NAME (COMPUTER): SEARCH (VAX)

DESCRIPTION:

SEARCH searches an ASCII text file for a specified character string.

LOCATION:

	Disk:[Directory]Name
Image:	DBB1:[FDYN.FDY05.SEARCH]SEARCH.EXE
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): SINGLE (VAX)

DESCRIPTION:

SINGLE converts unformatted IBM-S/360 single-precision data to internal VAX-11/780 representation.

CALLING SEQUENCE: FOUT = SINGLE(FINP)

Parameter	Type	Dimension	I/O	Description
FINP	R*4	1	I	REAL*4 value in IBM S/360 format. Before passing this value to SINGLE, the calling program must reverse the order of bytes in the long word
FOUT	R*4	1	O	REAL*4 value in VAX-11 internal format

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[GMAS]SINGLE.MAR

Sample of Use: None

LANGUAGE: MACRO

TYPE: Subroutine

DOCUMENTATION: "Guide for Converting Software From System 360 to VAX-11/780," memorandum from J. S. Watson, GSFC Code 580, April 1981

UTILITY NAME (COMPUTER): SLPPRINT (PDP)

DESCRIPTION:

SLPPRINT reformats a directory listing so that the files listed will be printed (titled and line numbered) on the line printer.

LOCATION:

	Disk:[Directory]Name
CMD File:	DB1:[204,12]SLP.CMD
Sample of Use:	None
Help File:	None

LANGUAGE: Monitor Console Routine (MCR)

TYPE: Command procedure

DOCUMENTATION: Prolog of SLPPRINT MAIN routine (location = DB1:[204,12]XXSLP.FPP)

UTILITY NAME (COMPUTER): SQUEEZ (PDP)

DESCRIPTION:

SQUEEZ removes anything in columns 73-80 and all trailing blanks from source code (for code taken from other computers).

LOCATION:

	Disk:[Directory]Name
Image:	DB1:[204,12]XXSQ.TSK
Sample of Use:	None
Help File:	None

LANGUAGE(S): Structured FORTRAN

TYPE: Program

DOCUMENTATION: Prolog of SQUEEZ MAIN routine
(location = DB1:[204,12]XXSQUEEZ.FPP)

UTILITY NAME (COMPUTER): SRD (PDP)

DESCRIPTION:

SRD alphabetizes a file directory, either by file name or by qualifier.

LOCATION:

	Disk:[Directory]Name
Image:	Installed as SRD command
Sample of Use:	None
Help File:	Available through PDP HELP utility

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): STRING (PDP)

DESCRIPTION:

STRING consists of an 11-routine library that performs character manipulation. The functions which can be performed are the following: concatenation, matching, text fragment searching, justification, locating the first occurrence or nonoccurrence of a character, and locating the last occurrence or nonoccurrence of a character.

LOCATION:

	Disk:[Directory]Name
Object Library:	DB1:[216,2]STRLIB.OLB
Sample of Use:	None

LANGUAGE(S): FORTRAN

TYPE: Library

DOCUMENTATION: None

UTILITY NAME (COMPUTER): STUFF(PDP)

DESCRIPTION:

STUFF is a FORTRAN-CALLABLE ROUTINE which will execute a single MCR command. Any valid MCR command can be executed, including indirect command files nested to any depth. (It should be noted that some arguments are not used. These arguments need not be coded, but their absence must be indicated by the appropriate commas.)

CALLING SEQUENCE: CALL STUFF(LUN,IBUF,ISIZE,IEFN,IRPI,ISTAT, ISW,IFNC,IUIC,IPAR,ITI)

Parameter	Type	Dimension	I/O	Description
LUN	I*2	1	-	Not used
IBUF	L*1	ISIZE	I	MCR command string (if terminated by a null byte, ISIZE is not required)
ISIZE	I*2	1	I	Length of IBUF in bytes (if IBUF is terminated by a null byte, ISIZE is not required)
IEFN	I*2	1	I	Event flag number to be set when the MCR command completes execution
IPRI	I*2	1	-	Not used
ISTAT	I*2	8	O	ISTAT(1) contains the completion status of the MCR command when it completes execution. ISTAT(2) through ISTAT(8) are required but not used.
ISW	I*2	1	O	SPAWN directive status word
IFNC	I*2	1	-	Not used
IUIC	I*2	1	-	Not used
IPAR	I*2	1	-	Not used
ITI	I*2	1	-	Not used

LOCATION:

Disk:[Directory]Name

Source Code: DB1:[216,2]STUFF.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): SUBMIT (VAX)

DESCRIPTION:

SUBMIT submits up to seven DCL commands, freeing the terminal when time-consuming commands are being executed.

LOCATION:

	Disk:[Directory]Name
COM File:	DBB1:[FDYN.FDY05]SUBMIT.COM DBB1:[FDYN.FDY05]SUB.COM
Sample of Use:	None
Help File:	None

LANGUAGE: Digital Command Language (DCL)

TYPE: Command procedure

DOCUMENTATION: None

UTILITY NAME (COMPUTER): TAP (PDP)

DESCRIPTION:

TAP performs disk-to-tape file transfer in EBCDIC or ASCII modes.

LOCATION:

	Disk:[Directory]Name
Image:	Installed as TAP command
Sample of Use:	None
Help File:	None

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): TAP (VAX)

DESCRIPTION:

TAP performs disk-to-tape file transfer in EBCDIC or ASCII modes. TAP operates from an MCR prompt and runs in compatibility mode.

LOCATION:

	Disk:[Directory]Name
Image:	Installed as MCR TAP command
Sample of Use:	None
Help File:	None

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): TAPECOPY (VAX)

DESCRIPTION:

TAPECOPY reads unlabeled magnetic tapes containing fixed-length records. Files are deblocked and written to disk.

LOCATION:

	Disk:[Directory]Name
Image:	DBB1:[FDYN.FDY05.TAPECOPY]TAPECOPY.EXE
Sample of Use:	None
Help File:	DBB1:[FDYN.FDY05.TAPECOPY]TAPECOPY.HLP

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: DBB1:[FDYN.FDY05.TAPECOPY]TAPECOPY.HLP

UTILITY NAME (COMPUTER): TEC (PDP)

DESCRIPTION:

TEC provides a text edit/correct feature. It is a character oriented editor available on all DEC machines.

LOCATION:

	Disk: Directory Name
Image:	Installed at TEC command
Sample of Use:	None
Help File:	(see TEC(VAX) description)

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: "TECO-11 Manual," Digital Equipment Corporation User's Society, November 1977
(also see the TEC(VAX) description)

UTILITY NAME (COMPUTER): TEC (VAX)

DESCRIPTION:

TEC provides a text edit/correct feature. It is a character-oriented editor available on all DEC machines.

LOCATION:

	Disk:[Directory]Name
Image:	Installed as MCR TEC command
Sample of Use:	None
Help File:	DBA0:[SYSMGR.DECUS]TECO.HLP

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: DBA0:[SYSMGR.DECUS]TECO.DOC

UTILITY NAME (COMPUTER): TIDY (VAX)

DESCRIPTION:

TIDY is a program that regularizes FORTRAN source code files. Statement numbers are put in ascending order, unreferenced statement numbers are removed, excessive blanks are removed, and excessive blank comments are deleted. Optionally, FORMAT statements can be collected at the end of each module, and DO loop and IF-THEN-ELSE-ENDIF constructs can be indented.

LOCATION:

	Disk:[Directory]Name
Image:	DBA0:[TOOLS.TOO02.TIDY]TIDY.EXE
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: M&DO OS/360 User's Guide, Vol. II
(GSFC publication, January 30, 1978)

UTILITY NAME (COMPUTER): TPU (PDP)

DESCRIPTION:

TPU provides limited tape handling capabilities, such as tape copying (tape-to-tape) and file directory production.

LOCATION:

	Disk:[Directory]Name
Image:	DB0:[1,54]TPU.TSK
Sample of Use:	None
Help File:	None

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): TRN (PDP)

DESCRIPTION:

TRN provides a tape-to-disk transfer capability for "card image" 80-byte records.

LOCATION:

	Disk:[Directory]Name
Image:	Installed as TRN command
Sample of Use:	None
Help File:	None

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): TRN (VAX)

DESCRIPTION:

TRN performs tape-to-disk transferring for "card image" 80-byte records. TRN operates from an MCR prompt and runs in compatibility mode.

LOCATION:

	Disk:[Directory]Name
Image:	Installed as MCR TRN command
Sample of Use:	None
Help File:	None

LANGUAGE(S): MACRO

TYPE: Program

DOCUMENTATION: None

UTILITY NAME (COMPUTER): URAND (PDP)

DESCRIPTION:

URAND produces random real numbers in the range (0.0,1.0). The method used in URAND is independent of the word size of the computer.

CALLING SEQUENCE: RANDX = URAND(IY)

Parameter	Type	Dimension	I/O	Description
IY	I*2	1	I	Arbitrary INTEGER value. (The value should be established before the first reference to URAND. IY should not be changed by the user program after the first reference.)
RANDX	R*4	1	O	REAL value in the range (0.0,1.0)

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[206,2]URAND.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: The Art of Computer Programming, Vol. II, "Seminumerical Algorithms," D. E. Knuth, 1969

UTILITY NAME (COMPUTER): UTCHAREQ (PDP)

DESCRIPTION:

UTCHAREQ provides for FORTRAN-callable string move and character comparison. (Entry points to UTCHAREQ are CHAREQ and MOVECR.)

CALLING SEQUENCE: LOGSW = CHAREQ(CHAR1,CHAR2)
CALL MOVECR(INBUFF,OUTBUFF,LENGTH)

Parameter	Type	Dimension	I/O	Description
CHAR1	L*1	1	I	Character number 1
CHAR2	L*1	1	I	Character number 2
LOGSW	L*1	1	I	Character comparison flag: = .TRUE. , if CHAR1 = CHAR2 = .FALSE., if CHAR1 ≠ CHAR2
INBUFF	Any	LENGTH	I	Input array field
OUTBUFF	Any	LENGTH	O	Output array field
LENGTH	I*2	1	I	Number of characters to move

LOCATION:

Disk:[Directory]Name

Source Code: DB1:[204,7]UTCHAREQ.MAC

Sample of Use: None

LANGUAGE: MACRO

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): UTGRAPH (PDP)

DESCRIPTION:

UTGRAPH generates a one-page Cartesian printer plot for any set of data with automatic scaling.

CALLING SEQUENCE: CALL GRAPH(X,Y,CHARS,N,IPR,MLINES,IOPT, XL,XH,YL,YH,LINES)

Parameter	Type	Dimension	I/O	Description
X	R*4	N	I	Array of X coordinates
Y	R*4	N	I	Array of Y coordinates
CHARS	L*1	N	I	Array of characters printed at each point
N	I*4	1	I	Number of points to be plotted (N ≥ 1)
IPR	I*4	1	I	FORTRAN unit number for printout
MLINES	I*4	1	I	Maximum number of lines desired in graph
IOPT	I*4	1	I	Option desired if two or more points fall in the same space: = 0, later characters override earlier ones = 1, insert a digit indicating number of points in each space (characters should not include +, -, or the numbers 2 through 9)
XL	R*4	1	I	Lower limit for the X values
XH	R*4	1	I	Upper limit for the X values
YL	R*4	1	I	Lower limit for the Y values
YH	R*4	1	I	Upper limit for the Y values
LINES	I*4	1	O	Actual number of lines printed (=1 indicates an error return; a message will print on unit IPR)

LOCATION:

	Disk:[Directory]Name
Source Code:	DB0:[201,1]UTGRAPH.FTN
Sample of Use:	None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Routine prolog

UTILITY NAME (COMPUTER): VAXEPH (VAX)

DESCRIPTION:

VAXEPH converts an IBM S/360 Goddard Trajectory Determination System (GTDS) ephemeris (EPHEM) tape to a VAX disk file with normal position and velocity units (i.e., kilometers and kilometers/second).

LOCATION:

	Disk:[Directory]Name
Image:	DBA0:[LNDSAT.PLAY]VAXEPH.EXE
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: None

C-2

UTILITY NAME (COMPUTER): VT100 (PDP)

DESCRIPTION:

VT100 is a collection of entry points which generates the appropriate escape sequence for cursor control on the DEC VT100 terminal. The entry points are all FORTRAN callable and provide such features as the following: clear all or part of the screen, set positive or negative video, position cursor (absolute or relative), and set margins.

LOCATION:

	Disk:[Directory]Name
Object Library:	DB0:[203,5]VT.OBJ
Sample of Use:	None

LANGUAGE(S): FORTRAN, MACRO

TYPE: Library

DOCUMENTATION: DECUS Document No. 11-424, "Routine to Drive the VT100," (location = DB0:[203,5]VT.TXT)

UTILITY NAME (COMPUTER): VT100 (VAX)

DESCRIPTION:

VT100 is a collection of subroutines which generates the appropriate escape sequences for cursor control on the DEC VT100 terminal. The routines are all FORTRAN callable and provide such features as the following: clear all or part of the screen, set positive or negative video, position cursor (absolute or relative), and set margins.

LOCATION:

	Disk:[Directory]Name
Object Library:	DBB1:[FDYN.FDY05.VT100]VT100.OLB
Sample of Use:	DBB1:[FDYN.FDY05.VT100]VTDEMO.FOR

LANGUAGE(S): FORTRAN, MACRO

TYPE: Library

DOCUMENTATION: DECUS Document No. 11-424, "Routine to Drive the VT100," which can be found on the STL VAX text file DBB1:[FDYN.FDY05.VT100]VT.TXT

UTILITY NAME (COMPUTER): WAIT (VAX)

DESCRIPTION:

WAIT generates a pause for a period of time specified by the input time and type. WAIT performs the same function as the WAIT system directive on the PDP-11/70.

CALLING SEQUENCE: CALL WAIT(ITIME,ITYPE,IRET)

Parameter	Type	Dimension	I/O	Description
ITIME	I*4	1	I	Time period to pause
ITYPE	I*4	1	I	Units of time: = 0, system ticks (nanoseconds) = 1, milliseconds = 2, seconds = 3, minutes = 4, hours
IRET	I*4	1	O	Return code: = 0, successful return = 1, input time greater than 1000 days

LOCATION:

Disk:[Directory]Name

Source Code: DBB1:[FDYN.FDY05.WAIT]WAIT.FOR

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: None

UTILITY NAME (COMPUTER): XEROX/XEROF (PDP)

DESCRIPTION:

XEROX/XEROXF writes standard label (EBCDIC) tapes for use with the Xerox 9700 system (M&DO xerox blue-book output). XEROX/XEROXF performs the necessary conversion and blocking.

XEROX assumes that the input file is list format (as produced by EDIT). XEROXF assumes that the input file contains FORTRAN carriage control characters.

LOCATION:

	Disk:[Directory]Name
Images:	DB1:[204,5]XEROX.TSK DB1:[204,5]XEROXF.TSK
Sample of Use:	None
Help File:	DB1:[204,5]XEROX.HLP

LANGUAGE(S): Pascal

TYPE: Program

DOCUMENTATION: DB1:[204,5]XEROX.HLP

UTILITY NAME (COMPUTER): XREF (VAX)

DESCRIPTION:

XREF produces a cross-reference listing of FORTRAN source code. The cross reference includes variable names, subroutine and function references, READ and WRITE statements, and statement labels.

LOCATION:

	Disk:[Directory]Name
Image:	DBB1:[FDYN.FDY05.XREF]XREF.EXE
Sample of Use:	None
Help File:	None

LANGUAGE(S): FORTRAN

TYPE: Program

DOCUMENTATION: "FORTRAN Cross-Reference Capability on PDP-11/70" (memo to the CSC Task 833 file)

UTILITY NAME (COMPUTER): YMD (PDP)

DESCRIPTION:

YMD converts a packed real variable of year, month, and day (YYMMDD.) into three separate integer variables for year, month, and day.

CALLING SEQUENCE: CALL YMD(T,IY,IM,ID)

Parameter	Type	Dimension	I/O	Description
T	R*8	1	I	Input date in packed format YYMMDD.
IY	I*4	1	O	Year from YY field of YYMMDD. input
IM	I*4	1	O	Month from MM field of YYMMDD. input
ID	I*4	1	O	Day from DD field of YYMMDD. input

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[207,1]YMD.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Routine prolog

Onboard Navigation Package (ONPAC) Simulator User's Guide (CSC/SD-81/6029)

UTILITY NAME (COMPUTER): YMD (VAX)

DESCRIPTION:

YMD converts a packed real variable of year, month, and day (YYMMDD.) into three separate integer variables for year, month, and day.

CALLING SEQUENCE: CALL YMD(T,IY,IM,ID)

Parameter	Type	Dimension	I/O	Description
T	R*8	1	I	Input date in packed format YYMMDD.
IY	I*4	1	O	Year from YY field of YYMMDD. input
IM	I*4	1	O	Month from MM field of YYMMDD. input
ID	I*4	1	O	Day from MM field of YYMMDD. input

LOCATION:

Disk:[Directory]Name

Source Code: DBA0:[LNDSAT.ONPAC]YMD.FOR

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Routine prolog

Onboard Navigation Package (ONPAC) Simulator User's Guide (CSC/SD-81/6029)

UTILITY NAME (COMPUTER): YMDHMS (PDP)

DESCRIPTION:

YMDHMS converts I*2 year, I*4 seconds, and I*4 microseconds to an R*8 packed date of the form YMDHMS.SS (year, month, day, hour, minutes, and seconds).

CALLING SEQUENCE: CALL YMDHMS(IYR,IYRSEC,IMSEC,YMDHMC)

Parameter	Type	Dimension	I/O	Description
IYR	I*2	1	I	Integer year
IYRSEC	I*4	1	I	Integer seconds of year
IMSEC	I*4	1	I	Integer microseconds of day
YMDHMC	R*8	1	O	Packed date of the form YMDHMS.SS

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[206,2]YMDHMS.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Routine prolog

UTILITY NAME (COMPUTER): YRSEC (PDP)

DESCRIPTION:

YRSEC converts a packed R*8 date of the form YMDHMS.ss to I*2 year, i*4 seconds, and I*4 microseconds. YRSEC is an entry to subroutine YMDHMS (another STL utility).

CALLING SEQUENCE: CALL YRSEC(YMDHMC,JYR,JYRSEC,JMSEC)

Parameter	Type	Dimension	I/O	Description
YMDHMC	R*8	1	I	Packed date of the form YMDHMS.SS
JYR	I*2	1	O	Integer year
JYRSEC	I*4	1	O	Integer seconds of year
JMSEC	I*4	1	O	Integer microseconds of day

LOCATION:

Disk:[Directory]Name

Source Code: DB0:[206,2]YMDHMS.FTN

Sample of Use: None

LANGUAGE: FORTRAN

TYPE: Subroutine

DOCUMENTATION: Routine prolog

REFERENCES

1. Computer Sciences Corporation, CSC/TM-81/6014, Software Engineering Laboratory (SEL) Compendium of Tools, W. J. Decker and E. J. Smith, February 1981

STL BIBLIOGRAPHY

Systems Technology Laboratory, STL-78-0001, System Description of the IMP-16C Microprocessor Orbit Determination Program, C. Shenitz, C. Rabbin, and G. Snyder, October 1978

--, STL-78-002, Landsat/NAVPAC System Description and User's Guide, S. R. Waligora, December 1978

--, STL-79-001, Intel 8080 Orbit Propagation Program System Description and User's Guide, C. Rabbin, April 1979

--, STL-79-002, Evaluation of the IMP-16 Microprocessor Orbit Determination System Filter, C. Shenitz, September 1979

--, STL-79-003, Multimission Modular Spacecraft Ground Support Software System (MMS/GSSS) MODCOMP Device and MAX IV Dependency Study, T. Weldon and M. McClellan, December 1979

--, STL-80-001, Orbit Determination Software Development for Microprocessor-Based Systems: Evaluation and Recommendations, C. M. Shenitz, July 1980

--, STL-80-002, The Two-Way TDRSS Observation Model for the LSI-11/23 Microcomputer, C. E. Goorevich, July 1980

--, STL-80-003, Automated Orbit Determination System (AODS) Requirements Definition and Analysis, S. R. Waligora, C. E. Goorevich, J. Teles, and R. S. Pajerski, September 1980

--, STL-80-004, Algorithms for Autonomous Star Identification, P. Gambardella, October 1980

--, STL-80-005, Autonomous Onboard Attitude Determination System Specifications and Requirements, M. D. Shuster, S. N. Ray, and L. Gunshol, December 1980

--, STL-81-001, Systems Technology Laboratory (STL) Library Methods and Procedures, W. J. Decker and P. D. Merwarth, September 1981

--, STL-81-002, Mathematical Specifications of the Onboard Navigation Package (ONPAC) Simulator (Revision 1), J. B. Dunham, February 1981

--, STL-81-003, Systems Technology Laboratory (STL) Compendium of Utilities, W. J. Decker, E. J. Smith, W. A. Taylor, P. D. Merwarth and M. E. Stark, July 1981

--, STL-81-004, Automated Orbit Determination System (AODS) Environment Simulator for Prototype Testing (ADEPT) System Description, S. R. Waligora, J. E. Fry, Jr., B. J. Prusiewicz, and G. N. Klitsch, August 1981

--, STL-81-005, Preliminary Automated Orbit Determination System (AODS)/AODS Environment Simulator for Prototype Testing (ADEPT) User's Guide, S. R. Waligora, Y. Ong, J. E. Fry, Jr. and B. J. Prusiewicz, September 1981

--, STL-82-001, GPSPAC/Landsat-D Interface (GLI) System Description, J. B. Dunham, H. M. Sielski, and W. T. Wallace, April 1982

--, STL-82-002, GPSPAC/Landsat-D Interface (GLI) System User's Guide, H. M. Sielski, J. B. Dunham, and P. D. Merwarth, March 1982

--, STL-82-003, Autonomous Attitude Determination System (AADS), Volume 1: System Description, K. A. Saralkar, Y. G. Frenkel, G. N. Klitsch, K. Liu and E. Lefferts, April 1982

--, STL-82-007, System Description for the Global Positioning Subsystem Experiment Package (GPSPAC) Experiment Data Preprocessor (GEDAP), P. D. Merwarth and J. F. Cook, June 1982