(12)

US Army Corps
of Engineers
Hydrologic Engineering Center

GENERALIZED COMPUTER PROGRAM

DSSMATH

Utility Program for Mathematical Manipulation of HEC-DSS Data

User's Manual

April 1992

CPD-64

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*
1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Distribution of this document is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) CPD-64	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Hydrologic Engineering Center 6b. OFFICE SYMBOL *(If applicable)* CEWRC-HEC	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS *(City, State, and ZIP Code)* 609 Second Street Davis, CA 95616-4687	7b. ADDRESS *(City, State, and ZIP Code)*
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL *(If applicable)*	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS *(City, State, and ZIP Code)*	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE *(Include Security Classification)*
DSSMATH - Utility Program for Mathematical Manipulation of HEC-DSS Data - User's Manual

12. PERSONAL AUTHOR(S)
CEWRC-HEC

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT *(Year, Month, Day)*	15. PAGE COUNT
Computer Program	FROM ________ TO ______	April 1992	55

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*
FIELD	GROUP	SUB-GROUP	Manipulation of HEC-DSS Data

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

User's Manual for program **"DSSMATH"**. **DSSMATH** enables mathematical manipulation of data stored in the Hydrologic Engineering Center's Data Storage System (HEC-DSS). The program provides capabilities for: arithmetic computations, transformations, such as stage to flow; screening; and estimation of missing or erroenous values. The program may be used in an automated batch environment for processing a real-time data stream, or it can be used interactively to perform ad hoc operations.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [✓ UNCLASSIFIED/UNLIMITED] ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION
22a. NAME OF RESPONSIBLE INDIVIDUAL AL MONTALVO	22b. TELEPHONE *(Include Area Code)* (916) 756-1104 22c. OFFICE SYMBOL CEWRC-HEC

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

DSSMATH

Utility Program for Mathematical Manipulation of HEC-DSS Data

User's Manual

April 1992

US Army Corps of Engineers
Hydrologic Engineering Center
609 Second Street
Davis, CA 95616-4687

(916) 756-1104

CPD-64

DSSMATH
Utility Program for
Mathematical Manipulation of HEC-DSS Data
User's Manual

Table of Contents

Chapter ... Page

Preface ... ii

1 Purpose ... 1

2 Description ... 1

3 Use

3.1 Running the Program ... 2
3.2 Use of Commands ... 2
3.2.1 Data Management ... 4
3.2.1.1 Program Memory ... 4
3.2.1.2 Data Retrieval and Storage ... 5
3.2.1.3 Memory Management Functions ... 5
3.2.2 Computations ... 6
3.2.2.1 Arithmetic Computations ... 6
3.2.2.2 Functions ... 7
3.2.3 Screening and Replacement ... 9
3.2.4 Plotting and Tabulation ... 9
3.2.5 Miscellaneous Commands ... 9
3.3 Contingency Processing ... 9

List of Figures

1 Example Input ... 4

List of Tables

1 Commands ... 3
2 Compute Functions ... 7

Appendices

A DSSMATH Commands
Command Syntax
Index of Commands
B Compute Functions
Index of Functions
C Programmer's Guide

PREFACE

Program **DSSMATH** enables mathematical manipulation of data stored in the Hydrologic Engineering Center's Data Storage System (HEC-DSS). The program provides capabilities for: arithmetic computations, transformations, such as stage to flow; screening; and estimation of missing or erroneous values. The program may be used in an automated batch environment for processing a real-time data stream, or it can be used interactively to perform ad hoc operations.

DSSMATH, previously known as WCCOMP, was developed in 1986 for the Sacramento District's Reservoir Control Section. WCCOMP was developed as a team effort between HEC personnel (Dennis Huff, Alfredo Montalvo and Jeff Houghten) and Sacramento personnel (Paul Pugner and Richard Jerome). Its primary function in 1986 was to provide the Sacramento District with the capability of doing mathematical manipulations on HEC-DSS time series data related to the operation of reservoir control systems. In October, 1991 WCCOMP was renamed DSSMATH and its scope of application broadened to encompass a wider range of mathematical manipulations of HEC-DSS data.

DSSMATH

1. Purpose

DSSMATH enables mathematical manipulation of data stored in the Hydrologic Engineering Center's Data Storage System (HEC-DSS). The program provides capabilities for: arithmetic computations, transformations, such as stage to flow; screening; and estimation of missing or erroneous values. The program may be used in an automated batch environment for processing a real-time data stream, or it can be used interactively to perform ad hoc operations.

2. Description

The present version of DSSMATH is essentially identical to a predecessor program WCCOMP. DSSMATH also encompasses selected routines from the program MATHPK. This user's manual is valid for all the different implementations (Harris, UNIX, and DOS).

DSSMATH supports regular, irregular-interval HEC-DSS time-series data and HEC-DSS paired function data. The standard memory configuration (Harris, UNIX, and DOS Lahey Extended) can accommodate 10 time series records of up to 5300 values each, 5 paired-data records each with 30 dependent variables with a maximum of 5000 values per record, and 15 scalars. Data may be accessed in several HEC-DSS files at the same time. In addition to the DOS Lahey Extended implementation, a cut down DOS version that runs under 640K is available. Its configuration is 5 time-series records consisting of 3180 values each and 3 paired functions consisting of 5 sets of dependent variables, with 250 values per paired function.

Time-series records and look-up tables are referenced in the various processing functions by user-assigned labels. The labels are one to six characters long and are assigned when the data is computed (CO command) or retrieved (GET command) from a HEC-DSS file. The HEC-DSS file name and data pathname are specified as parameters to the GET command (See Figure 1).

In either an interactive or in a batch setting, DSSMATH proceeds according to a sequence of instructions provided by the user. A typical sequence of instructions consists of a retrieval of data from the HEC-DSS file, a series of computations involving the data, and storage of results in a HEC-DSS file.

Computations in DSSMATH consist of arithmetic manipulations of time series data, such as multiplying a time series by a constant or another time series, and special functions that provide more complex and specific capabilities, such as rating table transformations.

Elementary screening functions flag data values internally to facilitate subsequent corrections. Replacements for flagged and missing values are provided by simple, linear estimation functions. Internal flags are used to assure that missing data do not erroneously affect computations. However, internal flags can not be read or written to HEC-DSS at present, and are not compatible with HEC-DSS Version 6 flag scheme.

The plotting capabilities of computer program DSPLAY have been incorporated into a special version of DSSMATH named DSSMATHD. These include control of the plotting parameters and options, such as grids and shading. The user assigned labels can be plotted by DSPLAY in addition to DSPLAY's standard displays of DSS data. The majority of the DSPLAY commands are available with the exception of commands that would conflict with DSSMATH commands. DSSMATHD is only available on the Harris Computer and at the present time there are no plans that this capability will be added to the UNIX and DOS versions. (Ref: HEC-4DSS, Users Guide and Utility Program Manuals)

The features of PREAD are provided in DSSMATH. PREAD facilitates efficient repetition of routine procedures. For example, a complex series of computations can be stored in a PREAD macro and started, for example, as a selection from a screen menu or a graphics tablet menu. (Ref: PREAD, Functions, Macros, Menus and Screens User Information)

3. Use

3.1 Running the Program

DSSMATH is initiated with: **DSSMATH [parameter] ...**

"parameter" has the form "key word=xxx" and may be one or more of the following specifications:

Key word	Default	Description
INPUT	*0	Terminal or batch job input file
OUTPUT	*3	Terminal or batch job output file
FUNFILE	MATHFUN	PREAD function file
MACFILE	MATHMAC	PREAD macro file
LOGFILE	W3	PREAD log file
SCNFILE	GENSCN	PREAD screen definition file

In the interactive environment, the default output is the user's terminal display. In the batch environment, the default output is the job output file.

3.2 Use of Commands

The processing of time-series data with DSSMATH, either interactively or through a batch input file, is directed with commands. Commands are specified with two-character mnemonics in columns' 1-2 of the input stream. The generalized form of the commands is:
XX[.options] [parameter 1] [parameter 2] ....

Commands typically require parameters. Parameters supply specific information about the objects or entities to be processed or specifications to control the process.

Options are single characters used to specify alternative ways of processing.

Input lines beginning with two asterisks (**) are used for notes or comments and are simply echoed in the output.

A list of commands is presented in Table 1. The commands are described in detail in the appendices A and B.

Table 1 Commands

Command	Description
**	Comment.
CATALOG	Display a catalog of a DSS file.
CLEAR	Remove a variable data label from memory.
COMPUTE	Perform a computation.
DIAG	Toggles DSS diagnostic trace output on and off.
DPATH	Display a selective catalog of a DSS file.
FINISH	Terminate and exit the program.
GET	Retrieve data from a DSS file.
HELP	List commands.
PLOT	Plot variables.
PUT	Store data in a DSS file.
SD	Set data descriptions.
SHOW	Displays selected internal information about the data variable denoted by specified variable label.
SMISSING	Set missing value indicators.
SP	Set data pathname.
STATUS	Display key program variables states or values.
TABULATE	Tabulate values of time-series or paired data.
TIME	Set a time-window for time series data.
$CO	Resume processing subsequent to error.
$AB	Abort (stop) processing subsequent to error.
DSPLAY	Initiate the DSPLAY program.

```
** Set a time window
TI T-2D T
**
** Get the "raw" data
GE STG=RAWDB:/SCIOTO/HIGH3/STAGE/01JUL1986/1HOUR/OBS/
**
** Get a look-up table for the station
GE TB=TABLEDB:/SCIOTO/HIGH3/STAGE-FLOW///USGS TABLE 6/
**
** Compute flows using the lookup table
COM FLW=RTABLE(STG,TB)
**
** Save the result
PUT FLW=MASTDB:/SCIOTO/HIGH3/FLOW/01JUL1986/1HOUR/COMP/
**
** Contingency checkpoint
$CONTINUE
**
** Clear program memories
CL  ALL
.
.
.
** Terminate
FI
```

Figure 1 Example Input

A typical command sequence retrieves a time series record, computes a dependent time series, and stores the result. An example input sequence, that computes flow values from stage values is shown in Figure 1.

3.2.1 Data Management

3.2.1.1 Program Memory.

The standard memory configuration can accommodate 10 time series records of up to 5300 values each, 5 paired-data records each with 30 dependent variables with a maximum of 5000 values per record, and 15 scalars. The DOS version configuration is limited to 5 time-series records of 3180 values each, 3 paired-data records each with 5 dependent variables with 250 values per record, and 15 scalars.

Time series records may be regular or irregular and all data times are stored. Data quality flags for each data value is also stored. The flags indicate four levels: 1) N for no flag; 2) E for estimated; 3) Q for questioned; and 4) M for undefined or missing. Time series records may be retrieved from files or may be computed in DSSMATH. However, currently, data quality flags may not be read or stored to DSS.

Paired data records include rating tables (ie., stage-flow) and polynomial coefficients. Paired-data records are usually retrieved from files, although at the present time only one compute function, "MATE", creates a paired-data record.

Scalars are variables that are results of assignments or computations. Storage for scalars is provided in order to use the scalar result of a computation in a subsequent computation.

Each record or variable in the program memory is assigned an alphanumeric label as it is retrieved or computed. The label is then used to refer to that time-series in further processing. The label may be one to six characters long. It must begin with an alphabetic character but may contain numeric characters. A label should not be the same as any of the commands or function names.

The most recently used time window, pathname, and DSS file name are kept in the program's memory. A maximum of five DSS files can be kept open on a continuous basis. Once a sixth DSS file is requested, the first DSS file opened is automatically closed and the new DSS file requested is opened. The number of DSS files that can be open on a continuous basis is limited to three for the 640K DOS version.

3.2.1.2 Data Retrieval and Storage. The commands "GE" (get) and "PU" (put) are used, respectively, to retrieve and store time series and paired function data. The location of the data in DSS files is determined by both a file name and a pathname (See Figure 1). Data may be retrieved and stored in different files as desired. A warning message is printed if the retrieval exceeds any storage limits. Options are provided with the command "PU" to control overwriting of previously stored data.

The time window command "TI" controls the span of the data retrieved with "GE" and, therefore, defines the period of time for processing. The time window must be defined before any processing can be accomplished. The time window should be defined carefully, since arithmetic computations and several of the compute functions check for concurrence of the data.

When regular-interval time series data are retrieved, the time window is temporarily adjusted, if necessary, to include data belonging to exact interval boundaries. Options with the retrieval command "GE" provide retrievals outside the range of the time window to facilitate use of irregular-interval time series data.

3.2.1.3 Memory Management Functions. The status command ("ST") may be used to show memory states and contents, including all variable label names and types (time series, paired function, or scalar), the current time window, and the most recently used DSS pathname and DSS filename. "ST" options will also show the pathnames, units and types of time series and paired-function variables and values assigned to scalars. The command "CL" releases memory spaces either globally or selectively, by label.

3.2.2 Computations. Computations result in the computation of values of a dependent time series, paired function, or scalar from a simple arithmetic operation or a more complex function of one or more time series, paired functions, or scalars. If the dependent time series or paired function variable is not in memory it is created with undefined pathname, type, and units. However, the type and units may be assigned by the function being used.

The computation of a dependen. ime series may be subjected to a test based on a logical IF condition that compares · lues of two variables or a variable and a constant. The variables may be concurrent time series or scalars. If the IF condition is not satisfied for a particular time, then the corresponding value of the dependent variable is unchanged. If the value was not previously defined, it will remain undefined.

In general, i. is left to the user to provide appropriate parameters in computations: for example, temperature values may be added to flows. However, arithmetic computations do require concurrent time series variables, and DSS data types or units are checked in certain compute functions. Usually, data types and units of a result must be specified explicitly as a separate processing step with the "SD" command.

3.2.2.1 Arithmetic Computations. An arithmetic computation of a time-series record consists of addition (+), subtraction (-), multiplication (*), division (/), or exponentiation (**) by a constant, previously defined scalar or time-series. Tho following are examples:

```
COM Y=X+2     add 2 to time series "X"
COM Y=X/A     "A" may be a scalar or time-series
```

One of the independent variables in an arithmetic computation may be redefined (ie., the dependent variable may be the same as one of the independent variables). However, if a scalar and a time series variable are the independent variables, the dependent variable is a time series.

When arithmetic computations involve two independent time-series, the dependent variable will consist of the arithmetic combination of the concurrent successive values of the two series. "Concurrent successive values" means equal numbers of values with the same times. The "TS1" or "TS4" compute functions are useful for aligning two parallel time series vectors in time. The "TS1" function generates a regular-interval time series by interpolation at regular intervals. The "TS4" function generates a time series interpolated to the times of a pattern time series.

Two time series involved in an arithmetic computation may have different units and types. Units and types of results must be explicitly defined by the user with the "SD" command.

3.2.2.2 Functions. Computations may also involve special functions that typically operate upon one or more time-series records and result in a scalar or new time–series record. The functions are summarized in Table 2 and described an appendix B.

Table 2 Compute Functions

Name	Description
ABS	Absolute function.
ACC	Running accumulation.
CMA	Centered moving average smoothing.
CORR	Compute correlation coefficients.
COS	Cosine trigonometric function.
COUNT	Count the number of valid and missing data.
DDT	Differences per unit time.
DECPAR	Decaying basin wetness parameter.
DIFF	Successive differences.
ESTLIN	Estimate values for missing data.
ESTPPT	Estimate values for missing precipitation.
FMA	Forward moving average.
GENTSR	Generate a regular interval time series.
INT	Truncate to whole numbers.
LOG	Natural log base "e".
LOG10	Log base 10.
MATE	Generate data pairs from two time series variables.
MAX	Maximum value in a time series.
MEAN	Mean value in a time series.
MIN	Minimum value in a time series.
MRG	Merge two time series.
MUSK	Muskingum routing function.
NINT	Round to nearest whole number.
OLY	Olympic smoothing.
PERCON	Period constants.

Table 2 (continued)

Name	Description
POLY	Polynomial transformation.
POLY2	Polynomial transformation with integral.
PULS	Modified Puls or Working R&D routing function.
QAC	Flow accumulator gage processor.
RND	Round off.
RTABLE	Rating table interpolation.
RTABLR	Reverse rating table interpolation.
RTABL2	Two variable rating table interpolation.
SCRN1	Screen for possible erroneous values based on maximum and minimum range.
SCRN2	Screen for erroneous data values based on a forward moving average maximum value.
SDEV	Compute the standard deviation of one independent variable.
SHIFT	Shift adjustment.
SIN	Sine trigonometric function.
SKEW	Compute the skew coefficient of one independent variable.
SQRT	Square root function.
SS	Straddle Stagger routing function.
SSW	Willmington District Straddle Stagger routing function.
TAN	Tangent trigonometric function.
TS1	Interpolate data at regular time intervals.
TS2	Period averages at regular intervals.
TS3	Period minimums and maximums at regular intervals.
TS4	Interpolated data at irregular intervals.
TSCYCL	Time Series Cyclic Analysis.
TSHIFT	Shift time series in time.
TTSR	Transform time series to regular.
TTSI	Transform time series to irregular.
1/X	Inverse function.

In addition to the functions summarized above, DSSMATH allows the sophisticated user who is proficient in FORTRAN, to create additional compute functions. See appendix C for instructions on creating additional functions.

Some computations result in a scalar value. Scalar values may also be assigned labels and used in subsequent processing, but cannot be retrieved or stored in DSS files.

3.2.3 Screening and Replacement

The screening and replacement capabilities of DSSMATH are implemented in the compute functions "SCRN1", "SCRN2", "ESTLIN", and "ESTPPT". Data values that exceed the specified limits are flagged internally to facilitate subsequent corrections with estimation functions or graphical editing.

3.2.4 Plotting and Tabulation

Plot and tabulation capabilities of DSSMATHD are nearly identical with those of the program DSPLAY. Time series and paired function data may be plotted or tabulated with the "PL" and "TA" commands, respectively. Commands that control the various plotting and tabulation options, such as grids and shading, are provided in DSSMATHD. Refer to the DSPLAY user's manual for additional instructions (HEC, HEC-DSS Users Guide and Utility Program Manuals). The plotting option is available only on the Harris version. The tabulate option is available on all versions of DSSMATH and is no longer dependent on the DSPLAY software.

3.2.5 Miscellaneous Commands

The DSS file catalog command "CA" is used for listing the pathnames of records in a DSS file. Data variable pathnames and descriptions (eg., units) can be specified using the "SP" and "SD" commands, respectively. On-line help is available through the "HE" command. The finish command, "FI", terminates processing.

3.3 Contingency Processing

Certain error conditions can render subsequent processing inappropriate. For example, if data cannot be retrieved, it cannot be processed. These errors, which simply result in warnings in the interactive environment, may be processed with recovery procedures when input are taken from an input file or a PREAD macro by specifying recovery checkpoints in the input.

An input line beginning with a dollar sign ($) is an error recovery checkpoint. "$CONTINUE" indicates where to resume processing when an error is encountered. "$ABORT" indicates the program is to stop if an error has been encountered and no $CONTINUE was found. If no checkpoints are found, processing stops.

Appendix A

DSSMATH Commands

Appendix A - DSSMATH Commands

Command Syntax

Brackets ([]) and ellipsis (...) symbols are notation used for describing commands and are not part of the actual command syntax.

UPPERCASE	Items in uppercase are required key words as shown.
lowercase	Items in lowercase are to be supplied by the user.
[ ]	Items within brackets are optional.
...	Items immediately preceding ellipsis may be repeated.

Commas and blanks are used to separate items and are interchangeable except as otherwise noted.

Periods are required when options are specified. No blanks should appear between commands and period or between periods and options.

Index of Commands

		Page
**	Comment	2
CATALOG	Display a catalog of a DSS file	2
CLEAR	Remove a variable data label from memory	3
COMPUTE	Perform a computation	3
DIAG	Diagnostic DSS trace	4
FINISH	Terminate and exit the program	5
GET	Retrieve data from a DSS file	5
HELP	List commands	6
PLOT	Plot variables	6
PUT	Store data in a DSS file	6
SD	Set data descriptions	7
SHOW	Display internal data variable information	8
SMISSING	Set missing value indicators	8
SP	Set data pathname	9
STATUS	Display key program variables states or values	9
TABULATE	Tabulate values of time-series or paired data	10
TIME	Set a time-window for time-series data	10
$CO	Resume processing subsequent to error	10
$AB	Abort (stop) processing subsequent to error	11
DSPLAY	Initiate the DSPLAY program	11

Name: ** **Comment**

Use: **[parameters] ...

Description: May be used to annotate input command streams.

Parameters:

text -	Any text up to 130 characters per line.

Example:

```
**   Set the time window
```

Name: **CATALOG** **Display a catalog of a DSS file**

Use: CA [.options] [parameters]

Description: Catalog (list) the records (pathnames) in a DSS file. The catalog is listed at the terminal one screen-full at a time. The numbers associated with each *record may be used in other commands to refer* to the record in lieu of a pathname.

Options:

None	(list old catalog in full mode)
N	(create new catalog)
A	(create abbreviated catalog)

Parameters:

None	Catalog last opened or referenced DSS filename.
filename:	Is the name of a DSS file.

Example:

```
CA.NA  MASDSS:
```

Name: CLEAR Remove a variable data label from memory

Use: CL [parameters]

Description: Clear releases memory slots. If the parameter 'ALL' is present, all slots are released. If any labels are specified, then only the memory slots represented by the labels are released.

Parameters:

None	No action is taken.
ALL	All variable labels are cleared and all data is initialized.
label...	Will clear specified variable labels.

Example:

```
CLEAR ALL
```

Name: COMPUTE *Perform a computationn*

Use: CO [parameters]

Description: DSS data header information are checked in some computations. If any of the independent variable values are flagged as missing then the computed value is missing. Data types, units of the dependent variable are generally undefined but may be set by some functions.

Parameters: [IF(x1 operator x2)] result=expression

"result" is a label for the resultant of the computation. The result may be a currently defined variable and may be used as an independent variable in subsequent computations.

"operator" is one of the following relationships:

LT less than	LE less than or equal
EQ equal	NE not equal
GT greater than	GE greater than or equal

"x1" and "x2"	are time series, scalars or constants. Time series should be concurrent with the operand.
"expression"	is either a simple arithmetic operation or a function. It may reference one or more labels for variables which have been previously computed or retrieved.

Name: DIAG Diagnostic DSS trace

Use: DI [parameters]

Description: Toggles diagnostic trace output on and off. Caution, this command can generate extensive output.

Parameters:

None	Diagnostic trace is off.
ON	Turn on diagnostic trace.
OFF	Turn off diagnostic trace.

Example:
```
DI  ON
```

Name: DPATH Display a selective catalog of a DSS file

Use: DP [.options] [parameters]

Description: Displays pathnames, tags and reference numbers from the catalog file (see CATALOG command). The DP command has the selective display capability which provides the option of selecting and listing only certain pathnames based on matching pathname parts.

Options:

None	(uses last referenced DSS file catalog)
N	(create new catalog)
A	(create abbreviated catalog)

Parameters:

None	Displays all pathnames.
filename:	Is the name of a DSS file.
A=..B=..etc	Selective catalog based on pathname parts.

Example:
```
DP MASDSS: B=RED CREEK C=FLOW
```

Name: FINISH Terminate and exit the program

Use: FI

Name: GET Retrieve data from a DSS file

Use: GE [options] [parameters]

Description:	Retrieves data from a DSS data file. The data may be time-series, or paired-function data (eg. rating tables or polynomial coefficients).

Options:

P	Includes the value just prior to the time window.
N	Includes the next value after the end of the time window.

Parameters:	label=[filename:]pathname
"label"	is an alphanumeric identifier for the data.
"filename"	is the DSS file name to use and it will be automatically opened if necessary. If omitted, the previously opened DSS file is used.
"pathname"	may be an explicit pathname or a pathname catalog number, or the previously specified pathname may be modified by specifying replacement pathname parts. If, for example, the previously defined pathname were:

/SCIOTO/CISG3/FLOW/01FEB1986/1HOUR/OBS/

and the following were specified:

B=HIGH3

then the new specification would become:

/SCIOTO/HIGH3/FLOW/01FEB1986/1HOUR/OBS/

Name: HELP List commands

Use: HE [parameters]

Description: Displays on-line documentation for a command. If no command is given, a list of commands is displayed. In order to get a list of the functions, enter "HE FUNCTION".

Parameters:

"name"	Command or function name for which a detailed description is needed.
FUNCTION	Will display a listing of the available DSSMATH functions.

Name: PLOT Plot variables

Use: PL [parameters]

Description: Plots the data represented by "label" using the integrated capabilities of DSPLAY. Up to seven labels may be plotted. Both time series and paired function data may be plotted.

Parameters:

"label"	is used to refer to the data within DSSMATH. Up to seven different labels may be specified.

Name: PUT Store data in a DSS file

Use: PU [options] [parameters]

Description: The PUT command is used to take the data in a variable label and store it in a DSS file if no parameters are specified, the data will be stored in the variables pathname if one has been given or in the last pathname defined.

Options:

A	All data in the time-series identified by label is written out to the DSS file "filename" and DSS record pathname, replacing any existing data.
R	Replaces existing data but it will not write out new DSS records consisting entirely of missing values.
M	Same as option **R**, except that existing data will not be replaced by a missing value.

Default is to replace only missing values; existing non-missing values will not be replaced.

Parameters: label=[filename:]pathname

"label"	identifies the time series to be written
"filename"	is the name of a DSS file to receive the data. If it is not specified, then the last previously referenced "filename" will be used. The DSS file will be automatically opened, if necessary.
"pathname"	is a definition of the pathname to use. The full pathname may be explicitly defined, or pathname parts may be used to modify the last previously defined pathname or the pathname associated with the label, if defined.

Name: SD Set data descriptions
Use: SD [parameters]

Description: Data descriptions are data items contained in DSS time series and paired function headers.

Parameters: label,parameter

"label"	is the DSSMATH label for data.
"parameter"	indicate which items to change and what the new values are. "parameter" has the form "item"="value", where "item" is TYPE or UNITS for time series data and TYPE, UNITS or LABELS for paired function data. The designation for "item" may be abbreviated, such as U= for UNITS= . "value" is an appropriate entry for the particular "item". For paired function data, enter one "value" for each curve in a series separated by commas. For example: U=FEET,CFS.

Name: SHOW Display internal data variable information

Use: SH [parameters]

Description: Displays selected internal information about the data variable denoted as a parameter to the SHOW command. Its main use is for debugging. However, this command is the only way to see the value of a scalar variable. For time-series it shows the data's array position, time (internal representation), time (external representation), value, and data quality flag. For paired data, it shows the DSS header information.

Parameters: label,parameter

None	No action is taken.
label	Label is the name of the data variable that will be displayed.
SCALARS	Display all scalar variables and their values.

Example:

```
SH  STG
```

Name: SMISSING Set missing value indicators

Use: SM [parameters]

Description: Missing value indicators are used to define the numeric values with which missing data is indicated. Up to 10 numbers can be specified. If no parameters are specified, default values -901.0 and -902.0 are used.
Caution: DSS will only accept the default values of -901.0 and -902.0 as valid, therefore be careful about writing data back to your DSS file when the missing value indicators have been changed from the default settings.

Parameters: "parameter" All valid numeric values.

Name: SP Set data pathname

Use: SP [parameters]

Description: The SP command is used to define or set the default pathname for a variable label name.

Parameters: label,parameter

"label" is the DSSMATH label for data.

"parameter" consists of either a pathname or a pathname part. If "parameter" is a part, it has the form "part"="value", where "part" is A, B, C, D, E, or F, and "value" is appropriate. If parts are specified, the resulting pathname for the data consists of the current default pathname modified by replacement of the specified parts.

Name: STATUS Display key program variables states or values

Use: ST [options] [parameters]

Description: The status command may be used to check on the state of key variables and to show data descriptions. Those functions are indicated with command options and parameters:

Options:

P	Display pathname for variable.
H	Display data descriptions (DSS header contents).

Parameters:

VAR	Show memory allocations, labels, and associated header information.
TIM	Show the current time window setting.
PA	Show the current pathname.
ALL	All of the above.

Name: TABULATE Tabulate values of time-series or paired data

Use: TA [options] [parameters]

Description: Tabulates the data represented by the variable label. Up to seven labels may be tabulated. Both time series or paired function data may be tabulated.

Options:

F	Send the output to the file specified on the execution line by the use of the parameter TAB=filename.

Parameters:

"label"	Name or label used for the data variable.

Name: TIME Set a time-window for time-series data

Use: TI [parameters]

Description: Starting and ending times and dates may be expressed in a variety of ways, including implicit and relative times and dates:

04MAR1983 0700	Complete, explicit expression.
D 0700	current date, time explicit
D	current date, time implicit (2400)
T	current date and time
T-2H	two hours ago
T-3D	three days ago
T-15M	15 minutes ago
D-30D	30 days ago, 2400 hrs
-D +D	expands the existing time window one day at each end

Name: $CO Resume processing subsequent to error

Use: $CO

Description: Recognized only in a batch job or when a PREAD macro is being used. Used to designate a point in the input command stream to resume processing when a preceding error has caused processing to terminate.

Name: $AB Abort (stop) processing subsequent to error

Use: $AB

Description: Recognized only in a batch job or when a PREAD macro is being used. Used to specify termination of the job if an error is encountered while processing according to a batch input stream.

Name: DSPLAY Initiate the DSPLAY program

Use: DSPLAY

Description: The DSPLAY command will initiate the graphics program DSPLAY. The user can either plot pathnames from the latest DSS file opened in DSSMATH or the user can use the OPEN command to open another DSS file. The majority of the regular DSPLAY commands are available. To exit the DSPLAY program, the user must enter the FINISH command. **(This command is only available on the Harris computer under program DSSMATHD)**

Appendix B

Compute Functions

Appendix B - Compute Functions

Index of Functions

		Page
ABS	Absolute function	3
ACC	Running accumulation	3
CORR	Compute correlation coefficients	3
COS	Cosine trigonometric function	4
COUNT	Count the number of valid and missing data values	4
CMA	Centered moving average smoothing	5
DDT	Differences per unit time	6
DECPAR	Decaying basin wetness parameter	6
DIFF	Successive differences	6
ESTLIN	Estimate values for missing data	7
ESTPPT	Estimate values for missing precipitation data	7
FMA	Forward moving average	7
GENTSR	Generate a regular interval time series	8
INT	Truncate to whole numbers	8
LOG	Natural log base "e"	8
LOG10	Log base 10	8
MATE	Generate data pairs from two time-series	9
MAX	Maximum value in a time series	9
MEAN	Mean value in a time series	9
MIN	Minimum value in a time series	9
MRG	Merge two time series	10
MUSK	Muskingum hydrologic routing	10
NINT	Round to nearest whole number	10
OLY	Olympic smoothing	11
PERCON	Period constants	11
POLY	Polynomial transformation	12
POLY2	Polynomial transformation with integral	12
PULS	Modified Puls or Working R&D routing function	12
QAC	Flow accumulator gage processor	13
RND	Round off	13
RTABLE	Rating table interpolation	14
RTABLR	Reverse rating table interpolation	14
RTABL2	Two-variable rating table interpolation	14
SCRN1	Screen for possible erroneous values based on maximum/minimum range	15
SCRN2	Screen for possible erroneous values	15
SDEV	Compute the standard deviation of one independent variable.	15
SHIFT	Shift adjustment	16
SIN	Sine trigonometric function	16
SQRT	Square root function	16

SS	Straddle Stagger routing function	17
SSW	Wilmington District Straddle Stagger routing function	17
TAN	Tangent trigonometric function	17
TS1	Interpolated data at regular intervals	18
TS2	Period averages at regular intervals	18
TS3	Period mins or maxs at regular intervals	18
TS4	Interpolated data at irregular intervals	19
TSCYCL	Time series cyclic analysis	19
TSHIFT	Shift time series in time	20
TTSR	Transform time series to regular	20
TTSI	Transform time series to irregular	21

Name:	**ABS**	**Absolute function**
Use:	CO TY=ABS(TX)	

Description: Compute the absolute value of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.

Name:	**ACC**	**Running accumulation**
Use:	CO TY=ACC(TX)	

Description: Compute a running accumulation of the values in TX and store the result in TY. If a TX value is undefined or a concurrent IF condition is not satisfied, the value of TX is not added to the accumulation and the corresponding TY remains the same as the previous TY. Units of TY are the same as those of TX. If TX is typed as PER-AVER or PER-CUM, TY will be typed INST-VAL or INST-CUM, respectively.

Name:	**CORR**	**Compute correlation coefficients**
Use:	CO TS=CORR(TX1,TX2,INDEX)	

Description: Compute the correlation coefficients, determination coefficients, and standard errors of regression between the two variables TX1 and TX2. TX1 and TX2 must be of the same type (either uniform time series or irregular time series) and contain the same number or rows or ordinates. Variable INDEX is used to set the scalar variable TS as follows:

Index	Description
1	Number of valid pairs for correlation.
2	Regression constant.
3	Regression coefficient.
4	Determination coefficient.
5	Standard error of regression.
6	Determination coefficient adjusted for degrees of freedom.
7	Standard error adjusted for degrees of freedom.

Name:	**COS**	**Cosine trigonometric function**
Use:	CO TY=COS(TX)	
Description:	Compute the cosine of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.	
Name:	**COUNT**	**Count the number of valid and missing data values**
Use:	CO TS=COUNT(TX,TYPE)	
Description:	This function is used to count the valid and missing values in time series variable TX and stores it in scalar variable TS. Possible values for TYPE are: "VALID" and "MISSING".	

Name:	**CMA**	**Centered moving average smoothing**

Use: CO TY=CMA(TX,NP[,CLEVEL])

Description: Compute a centered, moving average of NP values in TX and store in TY. NP must be odd and greater than 2. The default CLEVEL is "LEVEL3" and NP/2 values at the beginning and end of TY will be undefined. If a value in TX is undefined or a concurrent IF condition is not satisfied, then the resulting TY values are the averages of one less value. TX must be a regular-interval time-series and TY and TX cannot be the same. Useful for filtering instantaneous data containing high-frequency variations. Units and type are unchanged. The following levels are available:

LEVEL1 Only valid values will be averaged. Valid values that do not have all valid values within the averaging period number will be missing. Values at the start and end of the data that do not have enough valid values to average will be assigned missing values.

LEVEL2 Only valid values will be averaged. Valid values that do not have all valid values within the averaging period number will be missing. Valid values at the start and end of the data that do not have enough valid values to average will be averaged over a reduced number of values.

LEVEL3 (Default setting) All values will be averaged based on valid values within the averaging period number. If all values are missing for the period number, the value will be missing. Values at the start and end of the data that do not have enough valid values to average will be assigned missing values.

LEVEL4 All values will be averaged based on valid values within the averaging period number. If all values are missing for the period number, the value will be missing. Values at the start and end of the data that do not have valid values to average will be averaged based on a reduced number of values.

Note: Questionable and estimates flagged values are used in the computations.

Name: **DDT** **Differences per unit time**

Use: CO TY=DDT(TX)

Description: Computes the successive differences in TX per time period:

TY(t)=(TX(t)-TX(t-1))/DT

where DT is the time difference in days between t and t-1. If a value of TX is undefined TY is undefined. If a concurrent IF condition is not satisfied, TY is unchanged. TY and TX cannot be the same variable. TY is assigned the type 'PER-AVER.' The units of TY are undefined. An example of the use of DDT is the computation of reservoir inflow from outflow and the change in storage, where the change in storage is transformed to average flow volume per day by the function. The conversion factor for storage to flow is 0.50416 when the storage change is in ac-ft day.

Name: **DECPAR** **Decaying basin wetness parameter**

Use: CO TY=DECPAR(TY,TZ,R)

Description: Compute a time-series of parameters TY as a function of TZ and R:

TY(t)=R*TY(t-1)+TZ(t)

where R is the decay rate and TZ is precipitation. R is less than 1. The function extends TY: the first value in the series TY is used as the starting value, and any other TY are computed in the sequence. If the first value in TY is undefined, it is assumed to be zero. The first TZ value is ignored. TY and TZ must be regular-interval time series with identical times. R should be appropriate for the interval. The type and units of TY are unchanged. An IF condition has no effect.

Name: **DIFF** **Successive differences**

Use: CO TY=DIFF(TX)

Description: Compute the successive differences of TX and store in TY. TX and TY cannot be the same variable. TX must be of type INST-VAL or INST-CUM. If a value of TX is undefined, resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. The type and units of TY are undefined.

Name: **ESTLIN** **Estimate values for missing data**

Use: CO TY=ESTLIN(TX,NMAX,QFLAG)

Description: Linearly interpolate estimates for values in TX with flags equal or lesser in quality to QFLAG and place the results in TY. Do not estimate more that NMAX continuous missing values. Possible flags, in order of decreasing quality, are: Q = questionable and M = missing. An IF condition has no effect. TX must be a time-series, and TX and TY may be the same variable. Type and units of TY are the same as TX.

Name: **ESTPPT** **Estimate values for missing precipitation data**

Use: CO TY=ESTPPT(TX,NMAX,QFLAG)

Description: Linearly interpolate estimates for cumulative precipitation values in TX with quality flags equal to or lesser in quality to QFLAG and place the results in TY. Possible flags, in order of decreasing quality, are: Q = questionable and M = missing. If the values bracketing the missing period are increasing with time, do not estimate more that NMAX continuous missing values. If the values bracketing the missing period are equal, then estimate any number of missing values. If the values bracketing the missing period are decreasing with time, do not estimate any missing values. TX must be a data type of INST-CUM. An IF condition has no effect.

Name: **FMA** **Forward moving average**

Use: CO TY=FMA(TX,NP)

Description: Compute a moving average of the last NP values in TX and store in ıY. NP must greater than 2. NP values at the beginning of TY will be missing. If a value in TX is missing, the value is not used for computing TY, and the average is over one less value. At least 2 values of TX must be defined, else TY is missing. Useful for computing flow durations. Also, may be used to determine parameters for use in the SCR2 screening function. Units and type are unchanged.

Name: **GENTSR** **Generate a regular interval time series**

Use: CO TY=GENTSR(DT,TOFF,Y0,QFLAG)

Description: Generate a new, regular-interval time series TY with a time interval of DT, a time interval offset of TOFF, a constant value Y0, and all values internally flagged with QFLAG. TOFF is time from the beginning of the standard interval to the actual time of the data. DT and TOFF are expressed as units of time as nnT, where T is M for minutes, H for hours, and D for days. Possible flags are: N = none, M = missing, E = estimated, and Q = questionable. If Y0 is -901., then the flag is automatically M. Units and type must be set independently (use SDD). An IF condition has no effect.

Name: **INT** **Truncate to whole numbers**

Use: CO TY=INT(TX)

Description: Truncate to a whole number TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.

Name: **LOG** **Natural log base "e"**

Use: CO TY=LOG(TX)

Description: Compute the natural log base e of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. If a value in TX is 0 or negative, the value in TY will be set to missing.

Name: **LOG10** **Log base 10**

Use: CO TY=LOG10(TX)

Description: Compute the log base 10 of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. If a value in TX is 0 or negative, the value in TY will be set to missing.

Name:	**MATE**	**Generate data pairs from two time-series**

Use: CO PF=MATE(TX,TY,SORT/NOSORT)

Description: Derive a paired-function PF by pairing values in the time series TX and TY. TX and TY values must have identical times. (Functions TS1 and TS4 may be useful in conjunction with MATE). The paired data are sorted into ascending order of TX if SORT is specified. The units of PF are respectively the same as those of TX and TY. The types of PF are undefined. An IF condition has no effect.

Name:	**MAX**	**Maximum value in a time series**

Use: CO SY=MAX(TX)

Description: Find the maximum value in time-series TX and place it in scalar SY. Ignore missing values or values concurrent with an unsatisfied IF condition.

Name:	**MEAN**	**Mean value in a time series**

Use: CO SY=MEAN(TX)

Description: Compute the mean value in time-series TX and place the result in scalar SY. Ignore missing values or values concurrent with an unsatisfied IF condition.

Name: **MIN Minimum value in a time series**

Use: CO SY=MIN(TX)

Description: Find the minimum value in time-series TX and place the result in scalar SY. Ignore missing values or values concurrent with an unsatisfied IF condition.

Name:	**MRG**	**Merge two time series**
Use:	CO TY=MRG(TX,TZ,QFLAG)	
Description:	Merge TX with TZ. TY includes all values in TX and TZ, except where TX and TZ occur at the same time. In that case, the value for TX is used unless it is flagged with a quality equal to or less than QFLAG and TZ is flagged with a quality greater than QFLAG. Possible flags, in order of decreasing quality, are: N = no flag, E = estimated, Q = questionable and M = missing. An IF condition has no effect. The type and units of TY are undefined.	
Name:	**MUSK**	**Muskingum hydrologic routing**
Use:	CO TY=MUSK(TX,NR,K,X)	
Description:	Route the uniform time series variable TX by the Muskingum hydrologic routing method and store it in variable TY. The "IF" compute options has no effect on this function. The "K" parameter is the Muskingum "k" in hours, "X" parameter is the Muskingum "x" (range between 0 and .5), and "NR" is the number of routing subreaches.	
Name:	**NINT**	**Round to nearest whole number**
Use:	CO TY=NINT(TX)	
Description:	Round to nearest whole number TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.	

Name: **OLY** **Olympic smoothing**

Use: CO TY=OLY(TX,NP[CLEVEL])

Description: Compute a smoothed time-series from TX using the Olympic smoothing scheme: same as centered, moving average except the minimum and maximum values in the span NP are ignored. Place the result in TY. Units and type are unchanged. The following levels are available:

LEVEL1 Only valid values will be averaged. Valid values that do not have all valid values within the averaging period number will be missing. Values at the start and end of the data that do not have enough valid values to average will be assigned missing values.

LEVEL2 Only valid values will be averaged. Valid values that do not have all valid values within the averaging period number will be missing. Valid values at the start and end of the data that do not have enough valid values to average will be averaged over a reduced number of values.

LEVEL3 (Default setting) All values will be averaged based on valid values within the averaging period number. If all values are missing for the period number, the value will be missing. Values at the start and end of the data that do not have enough valid values to average will be assigned missing values.

LEVEL4 All values will be averaged based on valid values within the averaging period number. If all values are missing for the period number, the value will be missing. Values at the start and end of the data that do not have valid values to average will be averaged based on a reduced number of values.

Note: Questionable and estimates flagged values are used in the computations.

Name: **PERCON** **Period constants**

Use: CO TY=PERCON(TS,TX)

Description: Generates a series TY at times concurrent with TX and with values equal to the previous chronological TS value. If no previous TS is present, then TY values are undefined. TS and TX may be irregular- or regular-interval time series. An IF condition has no effect.

Name: **POLY** **Polynomial transformation**

Use: CO TY=POLY(TX,TP)

Description: Compute a polynomial transformation of TX using the polynomial coefficients TP. Store the result in TY. If a TX value is missing TY is undefined. If a concurrent IF condition is not satisfied, TY is unchanged. Units and type of TY are defined by TP. TP can be created with the utility DSSPD.

Name: **POLY2** **Polynomial transformation with integral**

Use: CO TY=POLY2(TX,TP)

Description: Compute a polynomial transformation of TX using the integral of the polynomial defined by coefficients TP. Store the result in TY. If a TX value is missing TY is undefined. If a concurrent IF condition is not satisfied, TY is unchanged. Units and type are unchanged. TP can be created with the utility DSSPD.

Name: **PULS** **Modified Puls or Working R&D routing function**

Use: CO TY=PULS(TX,PF,X,NR,STOR1,Q01)

Description: Route the uniform time series variable TX by the Modified Puls or Working R&D hydrologic routing method and store it in variable TY. The variable PF must have been previously defined in a GET command. It references a storage-discharge paired function relationship for the reach. Storage must be the first variable, discharge the second variable, and each variable must repeat only once. Variable X is the wedge coefficient (Muskingum X) for use in working R&D. Use 0.0 value to route by Modified Puls method. Variable NR is the number of routing reaches. STOR1 and Q01 are initial storage and flow respectively. Use a value of -1 for both STOR1 and Q01 in order to use the first flow value in TX and interpolate a corresponding storage value from PF. The IF compute command has no effect on this function.

Name:	**QAC**	**Flow accumulator gage processor**
Use:	CO TY=QAC(TX,TC)	

Description: Compute period-average flows from a flow accumulator type gage:

TY(t) = (TX(t)-TX(t-1))/(TC(t)-TC(t-1))

TX and TC are respectively, time series of the accumulated flow and the count. TX and TC values must occur at the same times. If corresponding values for TX and TC decrease from the previous period, then the accumulation is assumed to have reset to zero at the beginning of the interval. An IF condition has no effect. TY is assigned the type 'PER-AVER.'

Name:	**RND**	**Round off**
Use:	CO TY=RND(TX,NDIG,IPLAC)	

Description: Round off values to NDIG or IPLAC, whichever controls. NDIG is the number of digits to round to and can range from 1 to 8. IPLAC is a magnitude of 10 to which to round to: for example, -1 specifies rounding to one-tenth (0.1). *The number of digits shown* is never less than 1, however. The following example illustrates the effects on rounding if NDIG=3 and IPLACE=0 (ie., round to ones place):

	rounds	to
1445.1	"	1450.
144.51	"	145.
14.451	"	14.
1.4451	"	1.0

If TX is undefined, TY is undefined. If a concurrent IF condition is unsatisfied, TY is unchanged.

Name: **RTABLE** **Rating table interpolation**

Use: CO TY=RTABLE(TX,TB)

Description: Interpolate values for TX using table TB and store the result in TY. TB must be created using the program DSSPD (use /R option) with specific information in the header: type of interpolation, offset, shift, and datum. If the type of interpolation is LOGLOG, table x values are adjusted by subtracting the offset. The shift is added to and the datum subtracted from all incoming TX values. The header information in TB also defines the units of TY.

Name: **RTABLR** **Reverse rating table interpolation**

Use: CO TY=RTABLR(TX,TB)

Description: Interpolate values for TX using the reverse of table TB and store the result in TY. TB must be created using the program DSSPD (option /R) with specific information in the header: type of interpolation, offset, shift, and datum. If the type of interpolation is LOGLOG, table x values are adjusted by subtracting the offset. The shift is subtracted from and the datum added to all *resulting TX values.* The header information in TB also defines the units of TY.

Name: **RTABL2** **Two-variable rating table interpolation**

Use: CO TY=RTABL2(TX,TZ,TB)

Description: TY is a function of two independent variables TX and TZ. The functional relationship is specified by the table TB, which consists of sets of TX/TY pairs, each set corresponding to a TZ value. TB is created with the program DSSPD (option /R) with values for TZ specified as labels for the sets of TX/TY pairs. RTABL2 interpolates linearly in table TB. No extrapolation is done: if the TX or TY value are outside the range bounded by TB, TY is set to missing. TX and TZ must be concurrent time series. The header information in TB also defines the units of TY.

Name:	**SCRN1**	**Screen for possible erroneous values based on maximum/minimum range**

Use: CO TY=SCRN1(TX,XMIN,XMAX,MAXDEL,QFLAG)

Description: Flag any value in TX that falls outside the range XMIN - XMAX or exceeds the maximum change MAXDEL from the previous value. The maximum change comparison is done only when the consecutive values are not flagged. Possible values for QFLAG are: M = missing data or Q = questionable. The result is placed in TY.

Name:	**SCRN2**	**Screen for possible erroneous values based on forward moving average maximum**

Use: CO TY=SCRN2(TX,NPTS,MAXDEL,QFLAG)

Description: Flag any value in TX that exceeds the maximum change MAXDEL from the forward moving average of NPTS ending at the previous value. Missing values in TX are not counted in the moving average and the divisor of the average is less one for every missing value. Values which fail the screen are not counted either. At least 2 values must be defined else the moving average is undefined and the screen passes the relevant TX value. Possible flags are: M = missing data or Q = questionable. The result is placed in TY. SCRN2 is useful for detecting and removing spikes. The FMA function may be useful for determining appropriate NPTS and MAXDEL parameters.

Name:	**SDEV**	**Compute the standard deviation of one independent variable.**

Use: CO TS=SDEV(TX)

Description: This function is used to compute the standard deviation of one independent variable. TX can be either regular or irregular time series. TS is a scalar and is set to the computed standard deviation. This function requires a minimum of three valid values to compute.

Name: **SHIFT** **Shift adjustment**

Use: CO TY=SHIFT(TS,TX)

Description: Generate a time series of shift adjustments TY with times at TX and actual shifts TS, possibly at other times. TX and TY cannot be the same variable. TS values are interpolated at TX times. The interpolation is linear between TS values, except when TX time is greater then last TS time. Then, the last value of TS is held constant for the remaining TX times. If no previous TS value brackets TY, then TY is set to zero. An IF condition has no effect.

Name: **SIN** **Sine trigonometric function**

Use: CO TY=SIN(TX)

Description: Compute the sine of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.

Name: **SKEW** **Compute the skew coefficient of one independent variable**

Use: CO TS=SKEW(TX)

Description: This function is used to compute the skew coefficient of one independent variable. TX can be either regular or irregular time series. TS is a scalar and is set to the computed skew coefficient. This function requires a minimum of three valid values to compute.

Name: **SQRT** **Square root function**

Use: CO TY=SQRT(TX)

Description: Compute the square root of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. If a value in TX is negative, the value in TY will be set to missing.

Name: **SS** **Straddle Stagger routing function**

Use: CO TY=SS(TX,NAVG,LAG,NR)

Description: Route the uniform time series variable TX by the Straddle Stagger hydrologic routing method and store it in variable TY. Note: Variables TY and TX cannot be the same variable and variable TX is permanently modified by being lagged by the SS function. Variable NAVG is the number of ordinates to average. Variable LAG is the number of ordinates to lag hydrograph. Variable NR is the number of subreaches to use. The "IF" compute options has no (ect on this function.

Name: **SSW** **Wilmington District Straddle Stagger routing function**

Use: CO TY=SS(TX,NAVG,LAG,NR)

Description: Route the uniform time series variable TX by the Straddle Stagger hydrologic routing method and store it in variable TY. Note: Variables TY and TX cannot be the same variable. This function is similar to function SS except that LAG is usually zero and the result of averaging NAVG values is stored in the NAVGed value. Variable NAVG is the number of ordinates to average. Variable LAG is the number of ordinates to lag hydrograph. Variable NR is the number of subreaches to use. For example, if NAVG is 7, the results for ordinates 13 through 19 is stored in ordinate 19. For the same example, if LAG is set to 2, the result is stored in ordinate 21. The "IF" compute options has no effect on this function.

Name: **TAN** **Tangent trigonometric function**

Use: CO TY=TAN(TX)

Description: Compute the tangent of TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged.

Name:	**TS1** **Interpolated data at regular intervals**
Use:	CO TY=TS1(TX,DT,TOFF)
Description:	Derive a new, regular-interval time series TY from TX. TY will have a time span defined by the current time window, a time interval of DT, and a time interval offset of TOFF. TOFF is time from the beginning of the standard interval to the actual time of the data. DT and TOFF are expressed as units of time as nnT, where T is M for minutes, H for hours, and D for days. TX may be a regular- or irregular-interval time-series. Units and type are preserved. An IF condition has no effect.
Name:	**TS2** **Period averages at regular intervals**
Use:	CO TY=TS2(TX,DT,TOFF)
Description:	Derive a new, regular-interval time-series TY from the instantaneous values in TX with a time span defined by the current time window, a time interval of DT, and an interval offset of TOFF. TOFF is time from the beginning of the interval. DT and TOFF are expressed as units of time as nnT, where T is M for minutes, H for hours, and D for days. TOFF is time from the beginning of the standard interval to the actual time of the data. TX may be an irregular or regular time-series, but must be of type INST-VAL. TY will be typed PER-AVER. Example of use: deriving daily average flow from hourly observed values. An IF condition has no effect.
Name:	**TS3** **Period mins or maxs at regular intervals**
Use:	CO TY=TS3(TX,DT,TOFF,EXT)
Description:	Finds either the minima or maxima in TX at DT intervals, with interval offsets at TOFF, and puts the result in TY. EXT is used to indicate the extreme of interest: 'MIN' or 'MAX'. DT and TOFF are expressed as units of time as nnT, where T is M for minutes, H for hours, and D for days. TOFF is time from the beginning of the standard interval to the actual time of the data. TY is typed PER-EXTR. Example of use: finding daily minima and maxima from hourly instantaneous observations. An IF condition has no effect.

Name:	**TS4**	**Interpolated data at irregular intervals**

Use: CO TY=TS4(TX,TZ)

Description: Derive a new, irregular-interval time series TY from TX at the times for TZ (ie., TZ provides the time pattern). Units and type are preserved. An IF condition has no effect.

Name:	**TSCYCL**	**Time series cyclic analysis**

Use: CO XX=TSCYCL(TX,TREAT,STATFILE,DSSFILE)

Description: Derive a set of statistics from cyclic regular time series TX. TX must be regular data at a 1HOUR, 1DAY, or 1MONTH interval. The time series TX may be, for example, 30 years of 1DAY data. For each interval, (e.g., 1st day of the year, 2nd day of the year, ... , 365th day of the year), statistics would be determined. The 14 statistics determined for each interval are: maximum, minimum, average, standard deviation, 5%, 10%, 25%, 50% (median), 75%, 90%, 95% percentiles, date of maximum, date of minimum, and number of values processed. These results are written to a specified DSS file, and two text output files. The variable XX is ignored.

TREAT controls the treatment of newly generated statistic values when missing data occur in the interval. It must be expressed in either the form nn# or nn%. If nn# is used nn gives the number of missing data values in the interval that will be accepted in the computation of the new value. If nn% is used nn gives the percent of missing data values in the interval that will be accepted in the computation of the new value. If the criteria is met a data value will be generated, if not a missing value will occur.

STATFILE is the base name, not more than 6 characters, of two files that will be written containing the same data written to the DSS file. The first file will contain all of the statistics generated. It will be in the form of a wide table. Its name will be 'statfile.sts' or 'statfile.t' where 'statfile' is the portion of the name the user specifies. This file is wide, so it is hard to view on some systems. The second file is a subset of the first named 'statfile.sum', or 'statfile.s'. The second file is limited to 80 columns wide for easy viewing. The second file contains, maximums with their dates, minimums with their dates, average, and the 50% percentile(median) values.

DSSFILE is the DSS file into which the cyclic results will be written. The results will be, for example, a record of all the maximums for each day of the year over the 30 year period. Fourteen records will be added to the DSS file specified, each containing one of the derived statistics. The records will use the pseudo year 3000 for storing the statistics

Name: **TSHIFT** **Shift time series in time**

Use: CO TY=TSHIFT(TX,DT)

Description: Derive a time series TY by shifting times in TX by DT. DT is specified in the form nnT, where nn is the number of time units T and T may be M for minutes, H for hours, or D for days. Units and type are preserved. An IF condition has no effect.

Name: **TTSR** **Transform time series to regular**

Use: CO TY=TTSR(TX,DT,TOFF,FUNCT,TREAT,TYPE)

Description: Derive a new regular time series from existing time-series TX. TX may be regular or irregular. The new time series will be at a time interval of DT, and an interval offset of TOFF. DT and TOFF must each be expressed in the form nnT, where T may be MIN, HOUR, DAY, WEEK, MON, or YEAR (e.g. 1DAY). FUNCT is one of the following:

INT - Interpolation at end of interval
MAX - Maximum over interval
MIN - Minimum over interval
AVE - Average over interval
ACC - Accumulation over interval
ITG - Integration over interval
NUM - Number valid data over interval

TREAT controls the treatment of the new generated data value when missing data occur in the interval. It must be expressed in either the form nn# or nn%. If nn# is used nn gives the number of missing data values in the interval that will be accepted in the computation of the new value. If nn% is used nn gives the percent of missing data values in the interval that will be accepted in the computation of the new value. If the criteria is meet a data value will be generated, if not a missing value will occur.

TYPE provides the user control to override how the interpolation and processing of data occurs. TYPE must be one of the following: DEFAULT, INST-VAL, PER-AVER, or PER-CUM. If DEFAULT is used, processing depends on the data type stored in the DSS data record. Otherwise processing will be performed as if the data type were as given by TYPE. Data type INST-VAL considers the data to change linearly from the previous data value to the current data value over the interval. Data type PER-AVER considers the data to be constant at the current data value over the interval. Data type PER-CUM considers the data to increase from zero (0.0) up the current value over the interval.

Example: CO TY=TTSR(TX,1DAY,0MIN,AVE,10%,DEFAULT)

Name:	**TTSI**	**Transform time series to irregular**
Use:	CO TY=TTSI(TX,TZ,FUNCT,TREAT,TYPE)	
Description:	Derive a new irregular time series from existing time-series TX. TX may be regular or irregular. TZ is an existing time-series at the desired new spacing of TY. All other parameters are as shown for the function TTSR.	

Name:	**1/X**	**Inverse function**
Use:	CO TY=1/X(TX)	
Description:	Divide 1 by variable TX and store in TY. TX and TY can be the same variable. TX can be scalar, time series or paired data. If a value of TX is undefined resulting values of TY are undefined. If a concurrent IF condition is not satisfied, TY is unchanged. If a value in TX is zero, the value in TY will be set to missing.	

Appendix C

Programmer's Guide

Appendix C - Programmer's Guide

This appendix provides instructions on how a user, who is proficient in the use of the Fortran computer program language, can modify existing DSSMATH compute functions and create new user defined functions. The DSSMATH program has been structured to accommodate the relatively easy addition of new functions. The user modifies one standard DSSMATH subroutine "USRFUN", in which a user defined function is identified and then takes one of the existing function subroutines and modifies it to incorporate the new function. The most difficult task is the understanding of the large number of variables that DSSMATH uses to store all the information dealing with the function parameters and data values. All the variables are documented in the common blocks in which they reside.

The following instructions are given as an example in which a simple user defined function is created. The name of the new function will be **ABS(TX)**. The function will take the absolute value of all values in the **TX** variable.

Step 1. - Modify DSSMATH subroutine F.USRFUN

The shaded sections of code reflect the additions or modifications to the existing code to add the new user function **ABS**. Existing code is commented out; if it needs to be modified, it is duplicated and changes are made to the duplicated code.

```
      SUBROUTINE USRFUN (ISTAT)
C
C     PROVIDES A LINK WITH FUNCTIONS DEVELOPED LOCALLY
C     ****************************************************************
C
C     SUBROUTINE AUTHOR:  DENNIS HUFF
C     HYDROLOGIC ENGINEERING CENTER
C     DEVELOPMENT DATE:    5 FEB 88
C     REVISION LOG:
C     SUBROUTINE PARAMETERS:
C
C     ISTAT     -  (ISTAT = -1) ===>  SOMETHING WENT WRONG WHILE
C                      PROCESSING THE COMMAND LINE FOR THE FILE NAME.
C                  (ISTAT = 0 ) ===>  EVERYTHING IS OK.
$ADD C.IOCOM
$ADD C.CFUNCT
C
C     DETERMINE WHICH FUNCTION IS BEING USED
C
C     ****************************************************************
C     IF(CFUNEX(1:[illegible]).EQ.'[illegible]') THEN
C        CALL USRFUN(ISTAT)
      IF(CFUNEX(1:4).EQ.'ABS ') THEN
         CALL ABSFUN(ISTAT)
C
C     ****************************************************************
      ELSE
         ISTAT = -1
         WRITE(IFOUT,*)' ERROR - FUNCTION  ',CFUNEX(1:6),'  NOT VALID'
      ENDIF
C
      RETURN
      END
```

Step 2. - Modify DSSMATH subroutine F.ACC

The second step is to pick one of the existing DSSMATH functions and modify it to create the new function **ABS**. In the example below, a copy of the existing file **F.ACC** is modified to produce the new **F.ABSFUN** subroutine. **F.ACC** was chosen because it is a function that is similar to function **ABS**. The following is a listing of the DSSMATH function **ACC**. The common blocks have been turned on so the variable definitions are visible.

```
      [illegible]
C     [illegible]
C     [illegible]
C
      [illegible]
      SUBROUTINE ABSFUN(ISTAT)
C
C     COMPUTE ABSOLUTE OF TIME-SERIES VALUES
C
C        TY = ABS(TX)
C
C     ROUTINE CALLED FROM:
C
C     1.  SUBROUTINE FUNSUB
C
C     *****************************************************************
C
C     SUBROUTINE AUTHOR:  DENNIS HUFF
C     SUBROUTINE AUTHOR:  ALFREDO E. MONTALVO
C     HYDROLOGIC ENGINEERING CENTER
C     DAVIS CALIFORNIA
C     DEVELOPMENT DATE:   5  MAY 1987
C     DEVELOPMENT DATE:  21  JUNE 1990
C     REVISION LOG:
C
C
C
C     SUBROUTINE PARAMETERS:
C
C     -----OUTPUT-----
C
C     ISTAT    -  (ISTAT = -1) ===>  SOMETHING WENT WRONG WHILE
C                        PROCESSING
C                 (ISTAT = 0 ) ===>  EVERYTHING IS OK.
C
C     LOCAL VARIABLES:
C
C     NFPR     -  NUMBER OF PARAMETERS IN THE FUNCTION
C     ILGPTY   -  ARRAY CONTAINING THE LEGITIMATE PARAMETER TYPES
C     NITSA1   -  NUMBER OF PAIRS OF PARAMETERS TO CHECK FOR STARTING
C                 TIMES OF THE TIMES SERIES VARIABLES
C     ITSA1    -  INTEGER ARRAY USED TO POINT TO WHICH PARAMETERS
C                 ARE TO BE CHECKED AGAINST
C     NITSA2   -  NUMBER OF PAIRS OF PARAMETERS TO CHECK FOR DATA
C                 TIMES OF THE TIMES SERIES VARIABLES OCCURING AT
C                 THE SAME TIMES
C     ITSA2    -  INTEGER ARRAY USED TO POINT TO WHICH PARAMETERS
C                 ARE TO BE CHECKED AGAINST
C
C
```

```
C$ADO C.DSSDA1
C
C     ****************************************************************
C     THIS COMMON BLOCK CONTAINS VARIABLES DEALING WITH THE DSS DATA AND
C     VARIABLE NAMES BEING USED TO HOLD THIS DATA
C
      PARAMETER (NTS=10,NPF=5,NSCA=15,NPFC=30,NTSV=5000,NPFV=5000,
     & NPAT=NTS+NPF,NVAR=NTS+NPF+NSCA)
C
      COMMON /DSSDA1/ NHEADW(NPAT),HEADW(100,NPAT),NTYPE(NPAT),
     & NPNP(6,NPAT),NPATH(NPAT),ISTIME,ISDATE,IETIME,IEDATE,IHORIZ(NPAT)
     & ,TSDATE(NTSV,NTS),TSY(NTSV,NTS),PFX(NPFV,NPF),PFY(NPFV,NPF),
     & NCURVE(NPF),NDATA(NPAT),IPATH,SCALAR(NSCA),IQFLAG(NTSV,NTS)
C
      INTEGER HEADW
C
      COMMON /RDSSDA/ RBUFF
      REAL RBUFF(NTSV*2+100)
C
      COMMON /CDSSD1/ VARLBL,CPNP,CPATP,CSTIME,CSDATE,CETIME,CEDATE,
     & CUNITS,CTYPE,PATHNM,CARYLB
      CHARACTER VARLBL(NVAR)*6,CPNP(6,NPAT)*32,CPATP(6)*32,
     & CSTIME*4,CSDATE*7,CETIME*4,CEDATE*7,
     & CUNITS(2,NPAT)*8,CTYPE(2,NPAT)*8,PATHNM(NPAT)*80,
     & CARYLB(NPFC,NPF)*8
C
      COMMON /LDSSDA/ LTWSET
      LOGICAL LTWSET
C
C     DESCRIPTION OF VARIABLES
C
C     NPAT      -  MAXIMUM NUMBER OF PATHNAMES
C     NTS       -  MAXIMUM NUMBER OF TIME SERIES VECTORS
C     NPF       -  MAXIMUM NUMBER OF PAIRED FUNCTIONS
C     NSCA      -  MAXIMUM NUMBER OF SCALARS
C     NPFC      -  MAXIMUM NUMBER OF DEPENDENT VECTORS PER PAIRED
C                  FUNCTION
C     NTSV      -  MAXIMUM SIZE OF TIME SERIES VECTOR
C     NPFV      -  MAXIMUM SIZE OF PAIRED FUNCTION VECTOR
C     NVAR      -  MAXIMUM NUMBER OF USER VARIABLES
C     NHEADW()   -  LENGTH OF HEADER ARRAY
C     HEADW()    -  DSS DATA HEADER ARRAY
C     VARLBL()  -  USER VARIABLE LABELS
C                  FIRST NTS SLOTS POINT TO TS DATA
C                  NEXT NPF SLOTS POINT TO PF DATA
C                  LAST NSCA SLOTS POINT TO SCALARS
C     NTYPE()   -  TYPE OF DATA THAT IS HELD BY PATHNAME
C                  1 - TIME SERIES REGULAR TIME INTERVAL
C                  2 - TIME SERIES IRREGULAR TIME INTERVAL
C                  3 - PAIRED FUNCTION DATA
C                  4 - SCALAR VARIABLE
C                  5 - CHARACTER LABEL
C     IHORIZ    -  VARIABLE NUMBER TO APPEAR ON THE HORIZONTAL AXIS FOR
C                  PLOTTING (1 OR 2) - PAIRED DATA ONLY
C     IQFLAG()  -  INTEGER ARRAY USED TO INDICATE THE FOLLOWING FOR
C                  DATA VALUES STORED IN TSY() ARRAY.
C                  0 - NO QUALITY STATUS IMPLIED -
C                  1 - ESTIMATED VALUE
C                  2 - QUESTIONED VALUE
C                  3 - MISSING (UNDEFINED) VALUE FLAG
C     ISTIME    -  TIME WINDOW FOR STARTING TIME  (MINUTES)
C     IETIME    -  TIME WINDOW FOR ENDING TIME    (MINUTES)
C     ISDATE    -  TIME WINDOW FOR STARTING DATE  (DAY COUNT)
C     IEDATE    -  TIME WINDOW FOR ENDING DATE    (DAY COUNT)
C     CSTIME    -  CHARACTER STARTING TIME (24 HOUR TIME)
C     CETIME    -  CHARACTER ENDING TIME   (24 HOUR TIME)
C     CSDATE    -  CHARACTER STARTING DATE (MILITARY STYLE)
C     CEDATE    -  CHARACTER ENDING DATE   (MILITARY STYLE)
C     CPATP()   -  LAST PATHNAME ENTERED
C     IPATH     -  POINTER FROM CPATP TO CPNP
C     CPNP()    -  PATHNAME PARTS A-F
C     NPNP()    -  LENGTH OF PATHNAME PARTS A-F
C     CUNITS()  -  UNITS OF DSS DATA
C     CTYPE()   -  TYPE OF DSS DATA
C     PATHNM()  -  FULL PATHNAMES
C     NPATH()   -  LENGTH OF PATHNAMES
C     CARYLB()  -  LABELS FOR PAIRED FUNCTION DEPENDENT VECTORS
C     TSDATE()  -  TIME SERIES DATES (JULIAN DAYS WITH FRACTIONS)
C     TSY()     -  TIME SERIES DEPENDENT VARIABLE
```

```
C     PFX()      -  PAIRED FUNCTION INDEPENDENT VARIABLE                      $
C     PFY()      -  PAIRED FUNCTION DEPENDENT VARIABLES                       $
C     NCURVE()   -  NUMBER OF DEPENDENT VARIABLES PER PAIRED FUNCTION         $
C     NDATA()    -  NUMBER OF ORDINATES PER CURVE (PF OR TS)                  $
C     LTWSET     -  LOGICAL FLAG TO INDICATE IF THE TIME WINDOW IS SET        $
C     RBUFF      -  DSS WORK SPACE                                            $
C     SCALAR()   -  SCALAR VARIABLES                                          $
C                                                                             $
C     ****************************************************************        $
C                                                                             $
      COMMON /TBUFFR/ XTBUFF(NTSV),ITBUFX(NTSV)                               $
C                                                                             $
C     XTBUFF() -  TEMPORARY BUFFER SPACE USED TO STORE A REAL VARIABLE        $
C     ITBUFX() -  TEMPORARY BUFFER SPACE USED TO STORE AN INTEGER VARIABLE    $
C


C$ADD C.IOCOM
C                                                                             $
C     ****************************************************************        $
C     THIS COMMON BLOCK HOLD VARIABLES DEALING WITH INPUT AND OUTPUT          $
C     UNIT NUMBERS AND OPERATING ENVIRONMENT                                  $
C                                                                             $
      COMMON /IOCOM/ IFOUT,INPUT,IENVIR                                       $
C                                                                             $
C     VARIABLE DESCRIPTION                                                    $
C                                                                             $
C   INPUT   -  INPUT UNIT NUMBER                                              $
C   IFOUT   -  OUTPUT UNIT NUMBER                                             $
C   IENVIR  - =0 - INTERACTIVE EXECUTION                                      $
C             >0 - REAL TIME                                                  $
C             <0 - BATCH                                                      $
C     ****************************************************************        $
C                                                                             $

C$ADD C.CDEPND
C                                                                             $
C     ****************************************************************        $
C     THIS COMMON BLOCK CONTAINS VARIABLES DEALING WITH THE DEPENDENT         $
C     VARIABLE USED IN THE  COMPUTE COMMAND                                   $
C                                                                             $
      COMMON /CDEPND/ ICDPVA,ICDPTY,IDPVIF(5000)                              $
      COMMON /CCDEPN/ CDPVAR                                                  $
      CHARACTER CDPVAR*6                                                      $
C                                                                             $
C     DESCRIPTION OF VARIABLES                                                $
C     CDPVAR    -  CHARACTER VARIABLE USED TO STORE DEPENDENT VARIABLE        $
C                  NAME                                                       $
C     ICDPVA    -  INTEGER VARIABLE USED TO SPECIFY TWO THINGS:               $
C                  0 - DEPENDENT VARIABLE CDPVAR HAS NOT BEEN DEFINED         $
C                0 < - INDEX OF CDPVAR LOCATION OF ITS DATA                   $
C     ICDPTY    -  INTEGER VARIABLE  USED TO SPECIFY THE TYPE OF DATA         $
C                  STORED  IN DEPENDENT VARIABLE CDPVAR                       $
C     IDPVIF()  -   INTEGER ARRAY USED TO INDICATE IF THE DEPENDENT           $
C                   VARIABLE SHOULD BE PROCESSED BASED ON THE IF FUNCTION     $
C                   RESULTS                                                   $
C                   0 - PROCESS THE DATA VALUE                                $
C                   1 - DO NOT PROCESS THE DATA VALUE                         $
C                   2 - ONE OF THE IF FUNCTION PARAMETERS WAS MISSING         $
C                       AND THEREFORE DO NOT PROCESS THE DATA VALUE           $
C                                                                             $
C     ****************************************************************        $
C                                                                             $

C$ADD C.CFUNCT
C                                                                             $
C     ****************************************************************        $
C     THIS COMMON BLOCK CONTAINS VARIABLES DEALING WITH THE FUNCTIONS         $
C     USED IN THE COMPUTE COMMAND                                             $
C                                                                             $
      COMMON /CFUNCT/ ICFUNP(10),ICFPTY(10),ICFPSG(10),XCFUNP(10),            $
     . NFUNP,LFUNOP                                                           $
      LOGICAL LFUNOP                                                          $
      COMMON /CCFUNC/ CFUNEX,CFUNP                                            $
      CHARACTER*10 CFUNEX,CFUNP(10)                                           $
C     DESCRIPTION OF VARIABLES                                                $
C                                                                             $
C     CFUNEX    -  CHARACTER VARIABLE USED TO STORE THE FUNCTION NAME         $
C     CFUNP     -  CHARACTER VARIABLE ARRAY THAT HOLDS THE NAMES OF           $
```

```
C                 FUNCTION PARAMETERS                                          $
C       ICFUNP  -  INTEGER ARRAY USED FOR TWO PURPOSES                         $
C                  0 - INDICATES THAT FUNCTION PARAMETER IS A CONTANT          $
C                0 < - INDEX OF FUNCTION PARAMETER LOCATION                    $
C       ICFPTY  -  INTEGER ARRAY USED TO SPECIFY THE TYPE OF DATA STORED       $
C                   SEE DESCRIPTION OF NTYPE() VARIABLE IN DSSDA1 COMMON       $
C       ICFPSG  -  INTEGER ARRAY USED TO HOLD SIGN OF PARAMETER VARIABLE       $
C                  1 FOR POSITIVE AND -1 FOR NEGATIVE PARAMETER                $
C       LFUNOP  -  LOGICAL VARIABLE USED TO INDICATE IF THE  FUNCTION          $
C                  OPERATION IS BEING USED                                     $
C       NFUNP   -  NUMBER OF FUNCTION PARAMETERS                               $
C       XCFUNP  -  ARRAY USED FOR REAL NUMBER CONSTANT VALUE OF FUNCTION       $
C       ****************************************************************       $
C                                                                              $


C
C       SET SUBROUTINE PERCHK PARAMETER CHECKS FOR THIS FUNCTION
        DIMENSION ILGPTY(1),ITSA1(1),ITSA2(1)
        DATA NFPR,NITSA1,NITSA2 /1,0,0/
        DATA ILGPTY /12/
C
        CALL PERCHK( NFPR,ILGPTY,NITSA1,ITSA1,NITSA2,ITSA2,
       . ISTAT)
        IF(ISTAT.LT.0) GO TO 9000
C
        CALL STHDPV(1,ISTAT)
        IF(ISTAT.NE.0) GO TO 9000
C       [illegible]
C           [illegible]
C       [illegible]
C           [illegible]
C       [illegible]
        CTYPE(1,ICDPVA) = CTYPE(1,ICFUNP(1))
        CUNITS(1,ICDPVA)= CUNITS(1,ICFUNP(1))
C
C       PROCESS THE DATA
C       SUM = 0.
        DO 1000 I=1,NDATA(ICFUNP(1))
C          CHECK THE IF FUNCTION FLAG
        .  IF(IDPVIF(I).NE.0) THEN
C             TSY(I,ICDPVA)=SUM
              TSY(I,ICDPVA) = TSY(I,ICFUNP(1))
        .  .  IQFLAG(I,ICDPVA) = 0
C          CHECK TO SEE IF THE DATA IS MISSING
        .  ELSE IF(IQFLAG(I,ICFUNP(1)).GT.2} THEN
        .  .  IQFLAG(I,ICDPVA)=3
        .  .  TSY(I,ICDPVA)=-901
C          COMPUTE THE ACCUMULATION
        .  ELSE
C             SUM = SUM + TSY(I,ICFUNP(1))
C             TSY(I,ICDPVA)=SUM
              TSY(I,ICDPVA) = ABS(TSY(I,ICFUNP(1)))
        .  .  IQFLAG(I,ICDPVA)=0
        .  ENDIF
   1000 .  CONTINUE
C
        NDATA(ICDPVA)=NDATA(ICFUNP(1))
C
        NTYPE(ICDPVA) = NTYPE(ICFUNP(1))
        NHEADW(ICDPVA) = NHEADW(ICFUNP(1))
C
   9000 RETURN
        END
C
```

```
************
CAL-DSP.M    4/21/92
************

BOOTSTRAP
!R BEGIN

----------------------------------------------------------------------------

MACRO BEGIN
^z
!-FUNCTION
!SCREEN START
ENDMACRO

----------------------------------------------------------------------------

MACRO OPT $FILE
!PAGE
!JCL COPY $FILE.JOB NEXT.BAT > NUL
FINISH
ENDMACRO
============================================================================
----------------------------------------------------------------------------

MACRO TMP
!PAGE
!R PLT-TMP
!SCREEN PLOTHOLD
!PAGE
!SCREEN PLOT-OPT
ENDMACRO

----------------------------------------------------------------------------

MACRO PCP
!PAGE
!R PLT-PCP
!SCREEN PLOTHOLD
!PAGE
!SCREEN PLOT-OPT
ENDMACRO

----------------------------------------------------------------------------

MACRO GRD
!IF ( ^g .EQ. 'ON ' ) THEN
!TEACH g OFF
!ELSE
!TEACH g 'ON '
!ENDIF
!SCREEN PLOT-OPT
ENDMACRO

----------------------------------------------------------------------------

MACRO PLT-TMP
RESET.A
TI ^h ^u ^i ^v
US Davis, California
LE,1,Maximum Temperature
LE,2,Minimum Temperature
LE,3,Average Temperature
LE,4,Record Maximum Temp
LE,5,Record Minimum Temp
PA /^a/^b/TEMP-AIR MAX//^e/^f/
PA /^a/^b/TEMP-AIR MIN//^e/^f/
PA /^a/^b/TEMP-AIR AVE//^e/COMP/
PA /^a/^b/TEMP-AIR MAX-MAX//^e/OBS.1917-1986/
PA /^a/^b/TEMP-AIR MIN-MIN//^e/OBS.1917-1986/
YM,4,116,8.5,-2,A,Rec Max.
SW BLN BLN
!IF ( ^g .EQ. 'ON ' ) THEN
DG GR=MAJOR STYLE=DOTTED CROSS=ON
!ENDIF
DL CU=1 TY=2 COL=RED   SHADE=ON
DL CU=2 TY=2 COL=BLACK SHADE=ON
DL CU=3 TY=2 COL=GREEN SHADE=OFF
DL CU=4 TY=2 COL=YELLO SHADE=OFF
DL CU=5 TY=2 COL=CYAN  SHADE=OFF
TMARK T
PLOT
YM OFF
ENDMACRO


---------------------------------------------------------------------------


MACRO PLT-PCP
RESET.A
TI ^h ^u ^i ^v
US Davis, California
YL,1,Precip
YL,2,Cum. Precip
LE,1,Incremental Precipitation
LE,2,Cummulative Precipitation
LE,3,Normal Precipitation
LE,4,Percent of Normal Precip
PA /^a/^b/PRECIP-INC//^e/^f/
PA /^a/^b/PRECIP-CUM//^e/^f/
PA /^a/^b/PRECIP-CUM//^e/NORMAL/
PA /^a/^b/PRECIP-%NORMAL//^e/COMP/
SPLIT 80
SW TLI BRN BLN
!IF ( ^g .EQ. 'ON ' ) THEN
DG GR=MAJOR STYLE=DOTTED CROSS=ON
!ENDIF
DL CU=1 TYPE=1 COL=BLUE  SHADE=ON
DL CU=2 TYPE=2 COL=CYAN  SHADE=ON
DL CU=3 TYPE=2 COL=RED   SHADE=OFF
DL CU=4 TYPE=3 COL=YELLO SHADE=OFF
TMARK T
PLOT
ENDMACRO


---------------------------------------------------------------------------

MACRO EXIT
FINISH
ENDMACRO

---------------------------------------------------------------------------

MACRO SETT
!SCREEN TIME
!PAGE
!SCREEN PLOT-OPT
ENDMACRO

---------------------------------------------------------------------------

MACRO 1
!TEACH u 01OCT1991
!TEACH v T
!SCREEN TIME
ENDMACRO

---------------------------------------------------------------------------

MACRO 2
!TEACH u 01JAN1991
!TEACH v T
!SCREEN TIME
ENDMACRO

---------------------------------------------------------------------------

MACRO S
!TEACH l Start
!TEACH E 01JUN1991
!TEACH h 2400
!TEACH L u
!TEACH * 09
```

```
!SCREEN GTIME
!SCREEN TIME
ENDMACRO

---------------------------------------------------------------------------

MACRO E
!TEACH l End
!TEACH i 2400
!TEACH E 15JAN1992
!TEACH L v
!TEACH * 16
!SCREEN GTIME
!SCREEN TIME
ENDMACRO

---------------------------------------------------------------------------

MACRO RS
!TEACH l Start
!TEACH h ' '
!TEACH E T-30D
!TEACH L u
!TEACH * 09
!SCREEN GTIME
!SCREEN TIME
ENDMACRO

---------------------------------------------------------------------------

MACRO RE
!TEACH l End
!TEACH E T+5D
!TEACH i ' '
!TEACH L v
!TEACH * 16
!SCREEN GTIME
!SCREEN TIME
ENDMACRO

---------------------------------------------------------------------------

MACRO XTIME
!SCREEN PLOT-OPT
ENDMACRO

---------------------------------------------------------------------------

MACRO PLOT3
RESET.A
TI ^h ^u ^i ^v
US Davis, California
LE,1,Maximum Temperature
LE,2,Minimum Temperature
LE,3,Average Temperature
LE,4,Incremental Precip
PA /^a/^b/TEMP-AIR MAX//^e/^f/
PA /^a/^b/TEMP-AIR MIN//^e/^f/
PA /^a/^b/TEMP-AIR AVE//^e/COMP/
PA /^a/^b/PRECIP-INC//^e/^f/
SW BLN TRI
!IF ( ^g .EQ. 'ON ' ) THEN
DG GR=MAJOR STYLE=DOTTED CROSS=ON
!ENDIF
DL CU=1 TY=1 COL=RED   SHADE=ON
DL CU=2 TY=1 COL=BLACK SHADE=ON
DL CU=3 TY=1 COL=GREEN SHADE=OFF
DL CU=4 TY=2 COL=BLUE  SHADE=OFF
TMARK T
PLOT
!SC PLOTHOLD
!PAGE
!SC PLOT-OPT
ENDMACRO
```