HEC-DSS

User's Guide and

Utility Program Manuals

Overview

DSSUTL

DSPLAY

DSSMATH

REPGEN

DSSTS

DSSITS

DSSPD

DSSTXT

DWINDO

WATDSS

NWSDSS

PREAD

March 1995

Overview

Hydrologic Engineering Center
Data Storage System

March 1995

Hydrologic Engineering Center
U.S. Army Corps of Engineers
609 Second Street
Davis, California 95616-4687
(916) 756-1104

Acknowledgments

The development of Hydrologic Engineering Center Data Storage System (HEC-DSS) is due to the efforts of a number of individuals. The HEC-DSS was first conceived and designed in 1979 by Arthur F. Pabst, Mark G. Lewis, and Rochelle Barkin. Since that time, it has been upgraded and maintained by William J. Charley under the supervision of Arthur F. Pabst. Rochelle Barkin, Robert D. Carl, William J. Charley, Paul Ely, Dennis J. Huff, and Al E. Montalvo designed and implemented utility programs which support the HEC-DSS. Many others have provided ideas, interfaced application programs, and tested procedures.

Overview

Table of Contents

Appendices

Chapter 1

Overview Summary

The HEC Data Storage System (HEC-DSS or just DSS) stores data in a fashion convenient for inventory, retrieval, archiving and model use. The DSS was primarily designed for water resource applications. The user may interact with the data base through:

a) Utilities that allow entry, editing, and display of information.

b) Application programs that read from and write to the data base.

c) Library routines that can be incorporated in any program to access data base information.

The DSS provides a means for:

1) storing and maintaining data in a centralized location,

2) providing input to and storing output from application programs,

3) transferring data between application programs, and

4) displaying the data in graphs or tables.

(This page intentionally left blank)

Chapter 2

Introduction

2.1 Data Bases in General

The growth in use of data base systems is in response to the increasing, widespread use of computers in various applications. The ability to rapidly process large amounts of information has motivated the storage of information in a digital form. Essentially, data base systems have replaced written and printed storage of information.

In order to be effective, data base systems, like the paper-based systems they replace, are organized in a predictable format. Information is systematically labelled and stored to facilitate efficient and convenient reference and access.

Many kinds of information change in time. In order to maintain current information, data base systems provide the capability to add new information and revise or delete old information. This is referred to as updating the data base.

To achieve maximum utility, economy of storage, and ease of maintenance, information is kept in a central location for retrieval or updating by multiple users. Data base systems provide measures for minimizing information loss due to human error or computer system malfunctions. Use of a centralized storage location by multiple users also results in a need for measures to manage simultaneous requests for the same piece of information.

2.2 A Storage System for Water Resource Data

There is a substantial need for data base systems in water resources engineering and planning. Systematic, printed records of water resource data have been widely maintained since the beginning of the century, and the number of stations observed has greatly increased with time. Much of this enormous volume of printed data has been translated into digital storage medium, and it is now routine procedure to directly record observations in digital form. A data base system enables the storage of hydrologic data in a central location, and convenient retrieval capability.

Analysis programs are used to rapidly, repeatedly, and systematically evaluate long records of water resource data. In addition, programs often operate sequentially, with the output from one being the input to another. To be used effectively, these programs must be provided with the capability to conveniently store and retrieve data from a central storage location in a common format.

DSS was developed to meet the specific needs of applications in water resources studies. In order to efficiently manage large amounts of related data for practical applications, routines have been developed which organize and transfer data in blocks of continuous data. For example, each block of data for time series applications consists of a record of a time-dependent variable for a day or month or year. By storing data in this manner, an entire year of daily flows (or block of data) can be handled as a single element and accessed by one "search" of the data base. Other routines deal with tables of paired data, such as stage-discharge, stage-damage or discharge-frequency relationships.

Chapter 3

Concepts and Features

3.1 General

A conventional, general purpose data base system could be used for applications in water resources studies, but the large volumes of data typically involved make the generalized systems not the best choice for many applications. The generalized systems are often oriented towards business applications; they assume information is to be organized into collections of different attributes with a common relationship and store all information in character form. For example, the typical data base would efficiently store employee information - name, address, social security number, pay, etc., while the need in water resources may be to store 50 years of daily flows - quite different from the business application. Using an elemental business data base in the water resource application would require an excessive amount of computer time to search and convert character representation of numbers into real numbers.

HEC-DSS uses a block of sequential data as the basic unit of storage. This concept results in a more efficient access of time series or other uniquely related data. Each block contains a series of values of a single variable over a time span appropriate for most applications. The basic concept underlying the DSS is the organization of data into records of continuous, applications-related elements, as opposed to individually addressable data items. This approach is more efficient for water resources applications than that of a conventional data base system because it avoids the processing and storage overhead required to assemble an equivalent record from a conventional system.

DSS consists of a package of FORTRAN subroutines easily interfaced to computer programs, and a set of utility support routines to aid in interpreting and maintaining data in the data base. The subroutines and support programs retrieve (read) and store (write) data in "direct access" type computer files. Direct access files allow efficient retrieval and storage of blocks of data compared to sequential files. As many DSS files as needed may be created - the DSS does not limit the number of files. No "set-up" is required to generate a DSS file; the software does this automatically.

Data is stored in blocks, or records, within a file, and each record is identified by a unique name call a "pathname". Each time data is stored or retrieved from the file, its pathname must be given. Along with the data, information about the data (e.g., units) is stored in a "header array". The DSS automatically stores the name of the program writing the data, the number of times the data has been written to, and the last written date and time.

Through information contained in the pathname and stored in the header, the stored data is completely documented. That is, no additional information is required to identify it. The self documenting nature of the data base allows information to be recognized and understood months or years after it was stored.

3.2 Pathnames

DSS records are referenced by their pathnames. The pathname consists of up to 80 characters and is, by convention, separated into six parts. The parts are referenced by the characters A, B, C, D, E, and F, and are delimited by a slash "/", as follows:

```
/A/B/C/D/E/F/
```

For example, a brief description of the pathname for time series data is as follows:

PATHNAME PART	DESCRIPTION
A	River basin or project name
B	Location or gage Identifier
C	Data variable, e.g., FLOW, PRECIP
D	Starting date for block of data, such as 01JAN1981 for daily data in 1981
E	Time interval, e.g., 1DAY, 3HOUR, 1MON
F	Additional user-defined description to further define the data, e.g., PLAN A

A typical pathname might be:

```
/RED RIVER/BEND MARINA/FLOW/01JAN1975/1DAY/OBS/
```

The DSS software does not sequentially search through the DSS file for data, but uses its pathname to index its position within the file. This technique allows the rapid storage and retrieval of data from the file, regardless of its size.

3.3 Data Conventions

Data of most any type, using a pathname of any structure (up to 80 characters), can be stored by the DSS. To facilitate the ability of application and utility programs to work with and display data, standard record conventions were developed. These conventions define what should be contained in a pathname, how data is stored and what additional information is stored along with the data. For regular-interval time series data (e.g., hourly data), the conventions specify that data are stored in blocks of a standard length, uniform for that time interval, with a pathname that contains the date of the beginning of the block and the time interval. The conventions identify how a pathname for that data should be constructed. Conventions have been defined for regular and irregular interval time series data, paired (curve) data, and text (alphanumeric) data. Conventions for other types of data have been proposed.

Regular-interval time series data is data that occurs at a standard time interval (a list of intervals used in DSS is given in Appendix A). This data is divided into blocks, whose length depends on the time interval (for example, hourly data is stored with a block length of a month, while daily data is stored with a block length of a year). Only the date and time of the first piece of data for a block is stored; the times of the other data elements are implied by their location within the block. If a data element, or a set of elements, does not exist for a particular time, a missing data flag is placed in that element's location. Data quality flags may optionally be stored along with a regular-interval time series record. These flags can be used by several programs, and edited or displayed with the utility program DSSUTL.

Irregular-interval time series data does not have a constant time interval between values. This type of data is stored with a date/time stamp for each element. The user-selectable block size is based on the amount of data that is to be stored. For example, the user may select a block length of a month or a year. Because a date/time stamp is stored with each data element, approximately twice the amount of space is required compared with regular-interval time series data. Data quality flags may optionally be stored along with an irregular-interval time series record. These flags can be used by several programs, and edited or displayed with DSSUTL.

A convention for paired function data (or paired data) has been defined for data that generally defines a curve. It is used for rating tables, flow-frequency curves, and stage-damage curves, etc.. One paired data record may contain several curves within it, as long as it has a common set of ordinates. For example a stage-damage curve will contain a set of stages, and it may have associated residential damages, commercial damages, agricultural damages, etc.. However, a stage-damage curve and a stage-flow curve cannot be stored in the same record.

A text convention has been defined for lines of standard alpha-numeric characters (where a line begins with a carriage return and ends with a line feed character). This convention does not include characters that are not in a standard line format (such as those that would be used to generate a graphical display).

3.4 Multiple User Access

In order to maximize the effectiveness of a data base, several users should be able to use the same data file at the same time. For example, a flood forecast model may need to retrieve data from the file at the same time data elsewhere in the file is being updated. To do this, the data base must incorporate a method of handling multiple users of the same file at virtually the same time. A DSS multiple user scheme accomplishes this by a first come - first served approach. When a program requests to write to the file, the DSS software will queue that request and hold it until all prior requests are completed. Typical delays take only a fraction of a second and are not detectable by users.

3.5 File Protection

Measures to protect a file from program or system errors are an essential feature of a data base system. The DSS software accounts for this in its method of storage. An abort during the use of a DSS file usually results in no more than the loss of the data being currently written. As is true for all important information, the creation of a backup copy of a file is encouraged.

3.6 Application Program Interface

A systematic interface between application programs and DSS files is provided by a set of FORTRAN subroutines which are linked with application programs. A description of these subroutines and instructions for their use are presented in the HEC-DSS Programmer's Manual. Several of HEC's general application programs have been interfaced with DSS, allowing users to retrieve data for analysis or store results in a DSS file. This gives the user the capability of displaying and analyzing application program results by using the DSS utility programs. Using DSS to store program input and output helps avoid many of the cumbersome tasks associated with analyzing large amounts of data. These tasks include reformatting data, manually entering data, assembling large input files, etc. Programs using the DSS may interact with more than one DSS file simultaneously. Data can be retrieved from one file, analyzed, and then stored in another file.

3.7 Catalogs

Several DSS utility programs will generate a list of the pathnames in a DSS file and store that list in a "catalog" file. The catalog file is a list of the record pathnames in a DSS file, along with their last written date and time and the name of the program that wrote that record. The catalog is usually sorted alphabetically by pathname parts. Each pathname has a record tag and a reference number, either of which may be used in place of the pathname in several of the utility programs. The name given to the catalog file is the DSS file's name with a ".dsc" extension. An example of a catalog file is shown in Figure 1.

A catalog reference number is the sequential number of a pathname in the catalog file. These numbers are provided for quick interactive reference to a record from a utility program. When a number is given, the utility program sequentially searches the catalog file until it finds that number, then returns the associated pathname. Reference numbers are temporary; they may change each time the catalog is updated.

A record tag is a one to eight character semi-permanent record identifier that is not necessarily unique. It is also intended for quick interactive reference to a record from a utility program. Tags (which must begin with a non-numeric character), can be set by the user, or they can be set according to a scheme based on parts of the pathname. For example, a scheme might cause a data record of observed flow with a "B part" of NATP to have a tag of NATP-OF. The default tag is the letter "T" followed by the sequence number of the record. If a tag is not unique, only the pathname from the first matching tag will be used by the utility program. A file's tag scheme may be set, or records may have their tags' changed by the user with RT command in the program DSSUTL.

A special catalog, called a "condensed catalog", is useful primarily for time series data. In this type of catalog, pathname parts are listed in columns, and pathnames for time series data, which differ only by the date (D part), are referenced with just one line. Repeating parts are replaced by dashes for easier reading. The name given to the condensed catalog file is the name of the DSS file with a ".dsd" extension. An example of a condensed catalog is shown in Figure 2.

HEC-DSS Complete Catalog of Record Pathnames in File MASTDB.DSS

Catalog Created on Mar 6, 1990 at 15:24 File Created on Jan 8, 1990
Number of Records: 30 DSS Version 6-DA
Sort Order: ABCFED

Ref. Number	Tag	Program	Last Written Date	Last Written Time	Type	Vers	Data	Record Pathname
1	T3	DSSPD	06MAR90	15:22	RTS	1	366	/AMERICAN/AT FAIR OAKS/FLOW/01JAN1988/1DAY/OBS/
2	T4	DSSTS	06MAR90	15:22	RTS	2	365	/AMERICAN/AT FAIR OAKS/FLOW/01JAN1989/1DAY/OBS/
3	T11	DSSTS	06MAR90	15:22	RTS	3	365	/AMERICAN/BLUECANYON/PRECIP-INC/01JAN1987/1DAY/OBS/
4	T9	DSSTS	06MAR90	15:22	RTS	4	366	/AMERICAN/BLUECANYON/PRECIP-INC/01JAN1988/1DAY/OBS/
5	T12	DSSTS	06MAR90	15:22	RTS	1	365	/AMERICAN/BLUECANYON/PRECIP-INC/01JAN1989/1DAY/OBS/
6	T26	DSSTS	20FEB90	09:30	RTS	9	365	/AMERICAN/FOLSOM/ELEV/01JAN1987/1DAY/OBS/
7	T25	DSSTS	20FEB90	09:30	RTS	4	366	/AMERICAN/FOLSOM/ELEV/01JAN1988/1DAY/OBS/
8	T27	DSSTS	20FEB90	09:30	RTS	1	365	/AMERICAN/FOLSOM/ELEV/01JAN1989/1DAY/OBS/
9	T21	DSSTS	20FEB90	09:30	RTS	4	365	/AMERICAN/FOLSOM/EVAP-PAN/01JAN1986/1DAY/OBS/
10	T18	DSSTS	20FEB90	09:30	RTS	1	365	/AMERICAN/FOLSOM/EVAP-PAN/01JAN1987/1DAY/OBS/
11	T16	DSSTS	20FEB90	09:30	RTS	1	366	/AMERICAN/FOLSOM/EVAP-PAN/01JAN1988/1DAY/OBS/
12	T23	DSSTS	20FEB90	09:30	RTS	1	365	/AMERICAN/FOLSOM/FLOW-OUTLET/01JAN1987/1DAY/OBS/
13	T22	DSSTS	20FEB90	09:30	RTS	6	366	/AMERICAN/FOLSOM/FLOW-OUTLET/01JAN1988/1DAY/OBS/
14	T24	DSSTS	20FEB90	09:30	RTS	2	365	/AMERICAN/FOLSOM/FLOW-OUTLET/01JAN1989/1DAY/OBS/
15	T19	DSSTS	20FEB90	09:30	RTS	1	365	/AMERICAN/FOLSOM/FLOW-POWER/01JAN1987/1DAY/OBS/
16	T17	DSSTS	20FEB90	09:30	RTS	2	366	/AMERICAN/FOLSOM/FLOW-POWER/01JAN1988/1DAY/OBS/
17	T20	DSSTS	20FEB90	09:30	RTS	1	365	/AMERICAN/FOLSOM/FLOW-POWER/01JAN1989/1DAY/OBS/
18	T1	DSSPD	06MAR90	15:22	PD	1	52	/CALAVERAS/NEW HOGAN/STAGE-FLOW///USGS/
19	T2	DSSPD	06MAR90	15:22	PD	1	52	/CALAVERAS/NEW HOGAN/STAGE-FLOW/RT#13/30JUN1980/USGS
20	T6	DSSPD	06MAR90	15:22	PD	1	46	/DRY CR/NR GEYSERVILLE/STAGE-FLOW/RT#34/07MAR1986/USGS/
21	T5	DSSPD	06MAR90	15:22	PD	1	66	/DRY CR/NR GEYSERVILLE/STAGE-FLOW/RT #38/30SEP1987/USGS/
22	T8	DSSPD	06MAR90	15:22	PD	1	60	/DRY CR/NR LEMONCOVE/STAGE-FLOW///USGS/
23	T15	DSSPD	06MAR90	15:22	PD	1	14	/DRY CR/WARM SPRINGS/ELEV-AREA//01OCT1983/POLY/
24	T7	DSSPD	06MAR90	15:22	PD	1	62	/DRY CR/WARM SPRINGS/STAGE-FLOW///USGS/
25	T29	DSSTS	06MAR90	15:22	RTS	3	365	/FEATHER/QUINCY/PRECIP-INC/01JAN1982/1DAY/OBS/
26	T28	DSSTS	06MAR90	15:22	RTS	2	365	/FEATHER/QUINCY/PRECIP-INC/01JAN1983/1DAY/OBS/
27	T30	DSSTS	06MAR90	15:22	RTS	2	366	/FEATHER/QUINCY/PRECIP-INC/01JAN1984/1DAY/OBS/
28	T14	DSSTS	06MAR90	15:22	RTS	1	365	/FEATHER/SIERRAVILLE/PRECIP-INC/01JAN1986/1DAY/OBS/
29	T13	DSSTS	06MAR90	15:22	RTS	1	365	/FEATHER/SIERRAVILLE/PRECIP-INC/01JAN1987/1DAY/OBS/
30	T10	DSSTS	06MAR90	15:22	RTS	1	366	/FEATHER/SIERRAVILLE/PRECIP-INC/01JAN1988/1DAY/OBS/

Figure 1 - Catalog File

```
    HEC-DSS Condensed Catalog of Record Pathnames in File MASTDB.DSS

    Catalog Created on Mar 6, 1990 at 15:24     File Created on Jan 8, 1990
    Number of Records:     30                   DSS Version 6-DA
    Sort Order: ABCFED

Tag  A Part     B Part          C Part       F Part  E Part     D Part

T3   AMERICAN   AT FAIR OAKS    FLOW         OBS     1DAY       01JAN1988 - 01JAN1989
T11  - - - -    BLUE CANYON     PRECIP-INC   OBS     1DAY       01JAN1987 - 01JAN1989
T26  - - - -    FOLSOM          ELEV         OBS     1DAY       01JAN1987 - 01JAN1989
T21  - - - -    - - - - - - - - EVAP-PAN     OBS     1DAY       01JAN1986 - 01JAN1988
T23  - - - -    - - - - - - - - FLOW-OUTLET  OBS     1DAY       01JAN1987 - 01JAN1989
T19  - - - -    - - - - - - - - FLOW-POWER   OBS     1DAY       01JAN1987 - 01JAN1989
T1   CALAVERAS  NEW HOGAN       STAGE-FLOW   USGS    (null)     (null)
T2   - - - -    - - - - - - - - - - - - -    - - -   30JUN1980  RT #13
T6   DRY CR     NR GEYSERVILLE  STAGE-FLOW   USGS    07MAR1986  RT #34
T5   - - - -    - - - - - - - - - - - - -    - - -   30SEP1987  RT #38
T8   - - - -    NR LEMONCOVE    STAGE-FLOW   USGS    (null)     (null)
T15  - - - -    WARM SPRINGS    ELEV-AREA    POLY    01OCT1983  (null)
T7   - - - -    - - - - - - - - STAGE-FLOW   USGS    (null)     (null)
T29  FEATHER    QUINCY          PRECIP-INC   OBS     1DAY       01JAN1982 - 01JAN1984
T14  - - - -    SIERRAVILLE     PRECIP-INC   OBS     1DAY       01JAN1986 - 01JAN1988
```

Figure 2 - Condensed Catalog File

3.8 Utility Programs

Utility programs have been developed to manipulate or display data stored in a DSS file. Several of these programs are briefly described below. Users manuals for these products follow this overview, with exception of SHFDSS/DSSSHF and PIP which are found in the Water Control Software Data Acquisition and Flood Damage Analysis Package documents, respectively.

1) DSSUTL - General Utility Program

DSSUTL enables a user to copy, delete, rename, or edit data in a DSS file. DSSUTL may also be used in the transfer of data from one computer site to another. Files can be "squeezed" to reduce storage space by eliminating inactive space caused by record deletion. These functions are implemented by simple commands represented by a two letter code. A sorted and numbered inventory of pathnames in the file can be obtained by using the Catalog (CA) command. The tags or numbers from the list can then be used in lieu of the pathnames when using DSSUTL or other utility programs.

2) DSPLAY - Graphical Display Program

DSPLAY allows the user to easily display data from a DSS file in a tabular or graphical form. Data is retrieved from the DSS file by specifying the pathname (or tag or reference number) of the data the user wishes to display. The time command can be used to define the starting and ending times of the data, if different than the implied block times. Up to seven curves can be plotted on one graph, and a plot can be split into upper and lower graphs for variables with different units. The user can "window" in to enlarge portions of a plot. Graphical data can be edited using cross-hairs and cursor keys. Graph scaling, labels and legends are automatic, unless redefined by the user.

3) REPGEN - Report Format Program

REPGEN automates the production of routine reports in a variety of settings. REPGEN provides for the retrieval and presentation of data from a DSS file or a text file in a pre-specified, user-designed format. The format is the equivalent of a blank form onto which variable information is entered in designated locations. REPGEN could be used, for example, in a water control operations setting to automatically produce reports showing the current stage and flow at selected locations in a river basin.

4) DSSMATH - Mathematical Manipulation of DSS Data

DSSMATH provides a means of mathematically manipulating DSS data in a variety of ways. DSSMATH associates a variable name with a set of data from a DSS file. Data can be retrieved, manipulated by arithmetic operations or functions, and stored back into the same or a different DSS file. The user may add, subtract, multiply, divide or exponentiate uniform time series data. The user may compute many functions including the absolute value, truncate to whole numbers, round off numbers, remainder, square root, natural logarithm, minimum value, maximum value, mean value, sum of values, running accumulation, successive differences, standard deviation, skewness, correlation coefficients, standard error, interpolation by table lookup, derive different time series, perform Muskingum routing, straddle stagger routing, modified Puls routing, working R&D routing, and fill in missing data.

5) Data Entry Programs

A variety of utility programs are available for entering data into a DSS data base file. Some are designed to enter data from another data base. For example, WATDSS reads data from a file retrieved from the USGS WATSTORE system or a WATSTORE format file from a Compact Disk and enters it into a DSS file. Other programs are designed to enter data "manually" or in a generic form. For example, DSSTS is a prompt driven program for entering regular-interval time series data. A list of data entry programs is as follows:

Program	**Purpose**
DSSTS	Enter regular-interval time series data manually or from a file.
DSSITS	Enter irregular-interval time series data manually or from a file.
NWSDSS	Load daily and hourly climatological data (e.g., precipitation) from a National Weather Service tape or a NWS format file from a Compact Disk.
WATDSS	Load daily stream flow data from a WATSTORE file.
SHFDSS	Load data from a SHEF format (Standard Hydrometeorological Exchange Format). Companion program DSSSHF writes data from DSS into SHEF.
DWINDO	Edit or enter data in a full screen mode. This requires "forms" to be set up, and is intended for real-time data entry or editing.
DSSPD	Enter paired data manually or from a file.
PIP	Enter flood-damage paired data.
DSSTXT	Enter or display text data.

Chapter 4

Technical Information

4.1 Data Compression for Regular-Interval Time Series Data

Regular-interval time series data may be compressed with one or more of three methods. The method is selected by the user (or storing program) based on the kind of data that is to be stored.

The first method is a repeat counter scheme that flags duplicate values. It uses one bit per value to indicate a repeated value. It is often used for precipitation data, and can compress some data records by 97 percent. This compression method should not be used for data that is updated often (e.g., data stored in real-time), as it would likely cause the record to expand often and require excessive rewrites.

The second method compresses data by storing the differences between each value and the minimum value in the record. This is designed for data where the difference between the maximum value and the minimum value is not too large, and the precision of the values (the number of digits to the right of the decimal place) are known. The software uses the difference between the maximum and the minimum (or base) value, excluding missing data flags, and the precision number to determine the amount of space required for each number. If data is to be updated frequently with this method (e.g., entering real-time data in a master data base file), the base value and a storage size can be specified. This allows the software to update the data without having to recompute a base value and re-compress the record each time. Data compressed by this method are typically precipitation and stage values, and are compressed by 50 to 75 percent of their original size.

The third type of compression stores three significant digits for each value, and is often used to compress flow data. Data records compressed under with method are reduced in size by 50 percent.

The repeat method can be used in combination with the differences method or significant digits method. The differences method may not be used with the significant digits method.

4.2 DSS File Parameters

DSS file size parameters may be set for new files (or squeezed files) where the future size and type of the file is known. The DSS software has two methods to find a record from the pathname. In the first method, the software uses the pathname to look in a "dynamic" hash table for an index address, which points to a block containing the address of the actual data. This method is intended for where the file size may vary considerably (e.g., a data base file used for computations), or where the future size is not known. In the second method, the hash table is bypassed, and the blocks are pre-allocated when the DSS file is first generated. This typically saves one disk access when retrieving a record, but it can be very inefficient with disk space if the size used is incorrect. This method is intended for somewhat stable data bases that do not change in size frequently (e.g., a master data base file).

The hash table size, or the number of pre-allocated blocks, can also be controlled to optimize storage and retrieval of data according to the expected number of records in the file. It should be noted that any of the sizes will operate with any number of records, but an incorrect size will not be as efficient as the correct size. When a user squeezes a file with the program DSSUTL, the size parameter is automatically adjusted based on the number of records in the file at that time. The eight possible sizes are as follows:

Size Name	Ideal Number of Records	Target Range of Records
TINY	20	1-50
EXTRA-SMALL	50	1-200
SMALL	200	100-1,000
MEDIUM (default)	1,000	200-5,000
LARGE	4,000	1,000-10,000
EXTRA-LARGE	10,000	2,000-20,000
HUGE	20,000	5,000-50,000
EXTRA-HUGE	50,000	>25,000

These items are generally set by the experienced user who desires control over the optimization of data base files. The default parameters are sufficient for most users.

4.3 DSS on MS-DOS Personal Computers

The utility programs mentioned in this document are available for personal computers running the MS-DOS operation system. The use of these programs are similar to their use on other computers. In order to conserve memory, some of the utilities (DSPLAY and DSSUTL) are overlaid and must be placed in a directory that is set in the path (by the AUTOEXEC.BAT file). If the error "Cannot find program" is printed, the program's directory is probably not in the path (this error will occur even if you have typed the full directory when executing the program). It is recommended that the utility programs be installed in the directory "\HECEXE", and this directory name be added to the path in the AUTOEXEC.BAT file.

The graphics DSPLAY program uses device "drivers". Drivers provide a means of plotting on several different types of devices (monitors, pen plotters, printers, etc.), without having information about each device in the program. The specific drivers used by DSPLAY are loaded into memory at execution time. More information about drivers is included in the installation instructions provided with the DSPLAY program diskettes.

DSSUTL and DSPLAY sort catalog files by using either the DOS sort utility, or an external sort program that access extended memory (depending on the size of the catalog file). There must be at least 550 Kbytes of free conventional memory available in order to load the memory extender program and the sort program. If there is insufficient free conventional or extended memory, catalog files will not be sorted. The sort program (GNUSORT.EXE) and the memory extender (DOS4GW.EXE) must reside in the directory \HECEXE.

Several users can access a DSS file on a network server at the same time, if a multiple user access flag is set and the network server provides correct file locking capabilities. Unfortunately, it has been our experience that many network servers do not implement adequate file locking capabilities. Without file locking, the internal addressing tables in a DSS file can be overwritten when two users write new records to the file at approximately the same time. To prevent this damage, DSS file accessed from MS-DOS are opened by default in a single user mode. For network servers where file locking capabilities are implemented, a flag may be set to cause DSS files to be opened in a multi-user access mode. This flag is set by creating the file "\HECEXE\SUP\HECDSS.CFG", and entering the string "`MULTI-USER ACCESS`" on the first line (without quotes). This change will apply to all programs on that computer which access DSS.

4.4 DSS on UNIX Workstations

Most of the DSS utility programs are located in the directory "/usr/hec/bin" (or a link to this directory). Help and other supplemental files are located in the directory "/usr/hec/sup" (or linked directory). The graphics DSPLAY program will display plots on an X terminal device, or write graphics to a postscript file.

Multiple user accesses are accomplished by utilizing the system lock demon. If DSS is to be used in a NFS network mounted drives environment, compatible file-locking software (e.g., the lock demon) must be run both on the client and server machines to provide multiple user access to DSS files. This applies to drives mounted from another UNIX workstation, or from a DOS-PC. (Refer to section 4.3 for NFS access from MS-DOS.)

Appendix A

Data Conventions

Applications of hydrologic software generally involve use and generation of large numbers of data. The adoption of logical, consistent conventions for defined pathnames associated with the data can lead to (1) the capability to readily recognize from a pathname the type and other characteristics of information contained in a record, (2) a reduction in error associated with the storage and retrieval of information, and (3) the capability to readily adapt available macros, screens, etc. for working with standard types of data.

1. Time Series Conventions

This section covers the conventions for both regular-interval and irregular-interval time series data. There are four data "types" recognized by the DSS. They are:

Data Type	Example
`PER-AVER`	Monthly Flow
`PER-CUM`	Incremental Precipitation
`INST-VAL`	Stages
`INST-CUM`	Precipitation Mass Curve

Both regular and irregular interval time series record pathnames have the same A, B, C, and F parts. Data blocks are labeled with a six part pathname. The parts are referenced by the characters A, B, C, D, E, and F, and are separated by a slash "/", so that a pathname would look as follows:

/ A / B / C / D / E / F /

The conventions for the part labels are:

Part A - Group

The A part of the time series pathname describes the general group of the data. It may be a watershed name, a study name, etc., that would cause all associated records to be recognized as a group. The A part is not required.

Part B - Location

This part is the basic location identifier of the data. Generally the site name is used as the B part. A similar identifier, such as a project ID, USGS gage ID, or NWS station ID may be used. The B part is required.

When a hydrograph is routed from one location to another, the recommended B part is LOC1.LOC2, where LOC1 is the identifier <u>to</u> which the flow are routed, and LOC2 is an identifier for the location <u>from</u> which flows are routed. The second location (.LOC2) is optional. For example, it might be left out in situations where there is only one routed hydrograph for a location.

Part C - Parameter

The C part identifies the basic data parameter. A dash is used as a sub separator if further identification is needed. Additional information about the parameter, such as how it was obtained (e.g. "OBSERVED"), should be given in the F part. Examples of valid parameters are:

```
FLOW
ELEV
PH
PRECIP-INC
STAGE
TEMP-AIR
TEMP-WATER
```

Recommended C-parts for flows associated with stream locations are as follows:

C Part	Description
FLOW	Total flow.
FLOW-LOC	Local flow; that is, flow generated only from the subbasin that has an outlet at the specified location.
FLOW-CUM LOC	Cumulative local flow. This is the flow from all subbasins downstream from the nearest upstream reservoirs.
FLOW-COMB	Combined flow. This is the total flow minus the local flow.
FLOW-ROUT	Routed flow.
FLOW-DIVERT	Flow diverted out of the river at this location. The remaining flow would be labeled with a B-part of FLOW.
FLOW-mod	mod is a user-specified modifier for the flow. For example, FLOW-POWER, to designate a hydrograph from a power plant.

Recommended C parts associated with reservoirs are as follows:

C Part	Description
FLOW-IN	Reservoir inflow.
FLOW-OUT	Reservoir outflow.
FLOW-mod	mod is a user-specified modifier to flow associated with a reservoir, e.g. FLOW-IN1 for a component of reservoir inflow.
ELEV-RES	Reservoir elevation.
STORAGE	Reservoir storage.

Part D - Block Start Date

The D part of the pathname identifies the starting date of the data block. It is described for regular-interval and irregular-interval data in the sections that follow.

Part E - Time Interval or Block Length

The E part defines the time interval for regular-interval data, or the block length for irregular-interval data. It is further described in the sections that follow.

Part F - Descriptor

This part of the pathname is used to provide any additional information about the data. Its use may vary from application to application as appropriate, and may contain several additional qualifying pieces of information separated by a dash "-". If several forms of data exist, such as OBSERVED or FORECAST, and PLAN A or TEST 2, etc. they may be reflected in the F part. Generally the order of multi-descriptors of Part F should be from most to least significant.

A. Regular-Interval Time Series Conventions:

Regular-interval time series data is stored in "standard size" blocks whose length depends upon the time interval of the data. For example, daily time interval data are stored in blocks of one year (365 or 366 values), while monthly values are stored in blocks of ten years (120 values). If data does not exist for a portion of the full block, the missing values are set to the missing data flag "-901.0".

The starting and ending times of a block correspond to standard calendar conventions, e.g., for period average monthly data in the 1950's, the D part (date part) of the pathname would be `01JAN1950`, regardless of when the first valid data occurred (e.g., it could start in 1958). The 1960's block starts on `01JAN1960`.

Average period data values are stored at the end of the period over which the data is averaged. For example, daily average values are given a time label of 2400 hours for the appropriate day, average monthly values are labeled at 2400 hours on the last day of the month. If values occur for time other than the end-of-period time, that time offset is stored in the header array. For example, if daily average flow reading's are recorded at 6:00 am (i.e., the average flow from 6:01 am of the previous day to 6:00 am of the current day), then a offset of 360 (minutes) will be stored in the header array.

Part D - Block Start Date

The D part should be a 9-character military style date for the start of the standard data block (not necessarily the start of the first piece of data). Valid dates include `01JAN1982` for daily data, `01MAR1982` for hourly data, and `01JAN1900` for yearly data. Invalid dates include `01JAN82` (7 characters) and `14APR1982` for daily data (`14APR1982` is not the start of a standard daily block).

Part E - Time Interval

This part consists of an integer number together with an alphanumeric time interval specifying the regular data interval. Valid alpha entries are `MIN, HOUR, WEEK, MON`, and `YEAR`. The valid intervals and block lengths are:

Valid Data Intervals	Block Length
`1MIN, 2MIN, 3MIN, 4MIN, 5MIN, 10MIN`	One Day
`15MIN, 20MIN, 30MIN, 1HOUR, 2HOUR, 3HOUR, 4HOUR, 6HOUR, 8HOUR, 12HOUR`	One Month
`1DAY`	One Year
`1WEEK, 1MON`	One Decade
`1YEAR`	One Century

Examples of regular-interval pathnames are:

a. Daily USGS observed flow for station 0323150 for calendar year 1954 might be named:

```
/USGS/0323150/FLOW/01JAN1954/1DAY/OBS/
```

b. Six-hourly forecasted flow may have a pathname of:

```
/RED RIVER/DENISION/FLOW/01JUN1954/6HOUR/FORECAST/
```

c. Monthly elevations for Broken Bow Dam produced by application program HEC5 may be named:

```
/ARKANSAS/BROKEN BOW/ELEV/01JAN1950/1MON/PLAN 2R-RUN 5/
```

B. Irregular-Interval Time Series Conventions:

The irregular-interval time series conventions are similar to the regular-interval conventions, except that an explicit date and time is stored with each piece of data, whereas in regular-interval time series the date and time are implied by the location of the data within the block. Data is stored in variable length blocks (regular-interval data is stored in fixed length blocks). The block lengths are days, months, years, and decades.

The number of values that may be stored in one record is indefinite, although it is prudent to choose a size that will be less than 1000. The user selects the appropriate block length. For example, if the data to be stored occurred once every 1-2 hours, a monthly block would be appropriate. If data was recorded once or twice a day, use a yearly block. One would not want to store data that occurred 6 or more times a day in a yearly block (about 2200 values), because that may exceed dimension limits in some DSS programs.

All data is stored in variable length blocks that are incremented a set amount when necessary. Initial space for 100 data values is allocated. Additional increments are for 50 data values, unless otherwise set. (When the 101st data value is added to the record, a new record with a length of 150 values is written.)

Part D - Block Start Date

The D part should be a 9 character military style date for the first day of the standard data block (not necessarily the start of the first piece of data). For example, data stored in a daily block beginning on March 23, 1952 at 3:10 p.m. would have a D part of `23MAR1952`. If the same data were stored in a monthly block, the D part would be `01MAR1952`. The same data in a yearly block would have a D part of `01JAN1952`, and as a decade block, `01JAN1950`.

Part E - Block Length

The E part indicates the length of the time block, i.e. whether it is a day, a month, a year, or a decade. It consists of the letters "IR-" concatenated with the block length:

```
IR-DAY
IR-MONTH
IR-YEAR
IR-DECADE
```

It should be noted that the same data may be stored in blocks of different lengths. The DSS stores these as different records, and treats them as a completely different data set.

An example of a pathname for irregular-interval data is:

```
/SANTA ANA/PRADO/FLOW/01JAN1988/IR-MONTH/OBS/
```

2. Paired Data (Curve Data) Conventions

Paired data is a group of data that represents a two variable relationship. Typical examples are data that make up a curve (e.g., a rating table or a flow-frequency curve). Several curves may be stored in the same record if one of the variables is the same. For example several frequency-damage curves may be stored in the same record, where the curves may be residential, commercial, etc. A scale associated with the variable may be one of three types: linear, logarithmic, or probability. The pathname part identifiers are as follows:

Part A - Group

This part is for general grouping of data. It may be a study name, watershed name, etc., that would cause all records associated in a certain way to sort together, be copied together, delete together, or otherwise recognized as a group.

Part B - Location

This pathname part is the basic location identification of the data. It may be a control point, damage reach ID, station ID, or other identifier.

Part C - Parameters

Because paired data represents a relationship between 2 parameters, this part should contain the 2 parameter names separated by a hyphen (-). Examples of parameters are:

```
ELEV-DAMAGE
ELEV-FLOW
FREQ-FLOW
STATION-ELEV
```

In the above examples, `ELEV, FREQ`, and `STATION` are referred to as the first or independent variable, while `DAMAGE, FLOW` and `ELEV` are the second or dependent variable.

Part D - Optional Descriptor

This part of the pathname is used to provide any further descriptions of the data. It may vary from application to application as appropriate, and is often null.

Part E - Time Descriptor

This part of the pathname is used only if the paired data is representative of a specific point in time such as a 1995 forecast condition, or rating curve of `21MAR1981`.

Part F - General Descriptor

The F part identifies a unique descriptor of the data such as the situation, condition, or alternative plan name associated with the data. This part is included in labeling of data in some output utilities.

An example of a pathname for paired data is:

```
/ALLEGHENY/NATRONA/ELEV-DAMAGE//1980/FLOOD PROOF PLAN B/
```

3. Text Data Conventions

Text data is defined as generic alpha-numeric lines of text, where each line is preceded by a carriage return character and ends with a line feed character. It does not, at this time, include other types characters, such as those that would be used to create a graphical display. There are no definitive size limitations for a DSS text record, but it is recommended that a record contain no more than about 300 lines of text. There are no conventions set for the structure of the pathname. However, it is recommended that the pathname parts be labeled in a descending order of importance, and that the pathname indicate that the record contains text data and not one of the other types of data.

Text data may be entered manually or from a file with the program DSSTXT. DSSTXT, as well as DSSUTL, can also display (or place in a DSS file) text data.

Appendix B

Glossary of Terms

ASCII - American Standard Code Information Interchange. A standard "format" that defines most of the character symbols on a keyboard. This generally refers to "text" files that can be transferred and understood between different kinds of computers.

Batch (processing) - A means of executing programs or job control instructions in a background environment. Generally the program names or instructions are given in a file, and output is automatically sent to the printer.

Block - A related set or series of data stored together. For example, one year of daily flows would comprise one time series block (or record). The next year would comprise another block.

Blocked File - A type of computer storage file on disk. Blocked files are sequential in nature.

Binary File - A type of file that is usually created specifically for one kind of machine. Binary files are "non-text" files, and include such files as executable programs and object files.

Catalog - A listing of record pathnames and relevant information in a DSS file.

Condensed Catalog - A special catalog file for time series data. The condensed catalog shows pathname parts and the time span for data sets.

Data - The values stored or retrieved in a DSS file (e.g., daily stream gage readings).

Delimiter - A symbol separating a string of characters. A blank, a comma, or a slash may act as a delimiter.

Direct Access File - A type of computer storage file. Data in this type of file can be referenced by its position. For example, to retrieve data in "line" 1000, the computer can go directly to line 1000 to obtain the data, whereas in a sequential file, the computer has to read lines 1-999 first.

DSS (HEC-DSS) - Data Storage System; the general name of the software developed by the Hydrologic Engineering Center for storing and retrieving data.

File - A location on the computer's disk or tape where a user stores a set of information. Each file is given a unique name identifying it.

Header - A set of information stored with each DSS record. A record may have more than one header. Headers may describe the data in that record (e.g., the data's units), data compression information, and information that the user entered.

Job Control Language (JCL) - The means of a user giving instructions directly to the computer.

Interactive - Executing programs or JCL from a terminal or the keyboard.

Multiple User - A means of allowing several users to access and process data from the same DSS data file at the same time. A typical computer file will allow only one person to store and retrieve data at a time.

Option - A way of telling a program how to do something (not what to do it to). Used with commands in several DSS programs. Generally the option is separated from the command by a period.

Parameter - An instruction to a program telling it what to operate on. For example, in DSPLAY or DSSUTL, the pathname or pathname number following a tabulate command is the parameter. The parameter is generally separated from the command by a comma or blank.

Pathname - An unique identifier label for a block of DSS data. The pathname may consist of up to 80 characters and is, by convention, separated into six parts with a slash "/" delimiter. When accessing data in a DSS file, the data's pathname must be given.

Pathname Part - A conventional pathname is separated into six parts, separated by slashes. Each part is identified by one of the letters A, B, C, D, E, or F. The part labels are given in the section "Data Conventions".

Random File - A file allowing reading and writing by multiple users at the same time.

Record - For DSS usage, a record is equivalent to a block of data, i.e., all data stored under one pathname is called a record. For example, a year of daily flows would compromise one time series record.

Reference Number - A listing number for a pathname from the catalog file. The reference number can often be used in place of the complete pathname in DSSUTL and DSPLAY. These numbers may change each time a new catalog is created.

Routine (subroutine) - A set of instructions called by a program. All DSS subroutines are identified by a "Z" as the first character of their name.

Sequential Access File - A type of computer storage file. Data in this type of file must be obtained sequentially. For example, to retrieve data in "line" 1000, lines 1-999 must be read first.

Tag - A one to eight character "semi-permanent" record identifier for interactive use. A record's tag may often be used in place of the complete pathname in DSSUTL and DSPLAY. A tag may be set by the user or by a program.

Version (DSS software) - The software version indicates which group of DSS software created the file. The version appears at the top of the catalog, and is a number followed by two letters (e.g., 6-EA). The number changes on a substantial overhaul of the software. The first letter changes for each major change in the software, and the second letter for every minor change. DSS files with different version numbers are not compatible with each other, while files only with different version letters are compatible.

Word (computer word) - The basic storage element of the computer system. Any number stored (in a non-compacted form) takes at least one word. The length of a word is usually measured in bytes (the storage required for one character). On the PC, one word requires 2 bytes, while on the UNIX computers, 4 bytes comprise one word.

ZCLOSE - A DSS software subroutine that updates and releases a DSS file from a program.

ZOPEN - A DSS software subroutine that attaches a DSS file to a program and obtains basic information from the file, allowing reading and writing to proceed.

ZREAD - A DSS software subroutine that retrieves data from a DSS file based on the pathname used. When data is retrieved, a message is printed giving the pathname of the record read (a program can suppress this message).

ZWRITE - A DSS software subroutine that stores data in a DSS file based on the pathname used. ZWRITE prints out a message indicating which record was written and its version number.