

SPERRY UNIVAC
Operating System/3 (OS/3)
Introduction to
Data Utilities

This document contains the latest information available at the time of publication. However, Sperry Univac reserves the right to modify or revise its contents. To ensure that you have the most recent information, contact your local Sperry Univac representative.

Sperry Univac is a division of Sperry Rand Corporation.

AccuScan, FASTRAND, PAGEWRITER, SPERRY UNIVAC, UNISCOPE, UNISERVO, and UNIVAC are trademarks of the Sperry Rand Corporation.

preface

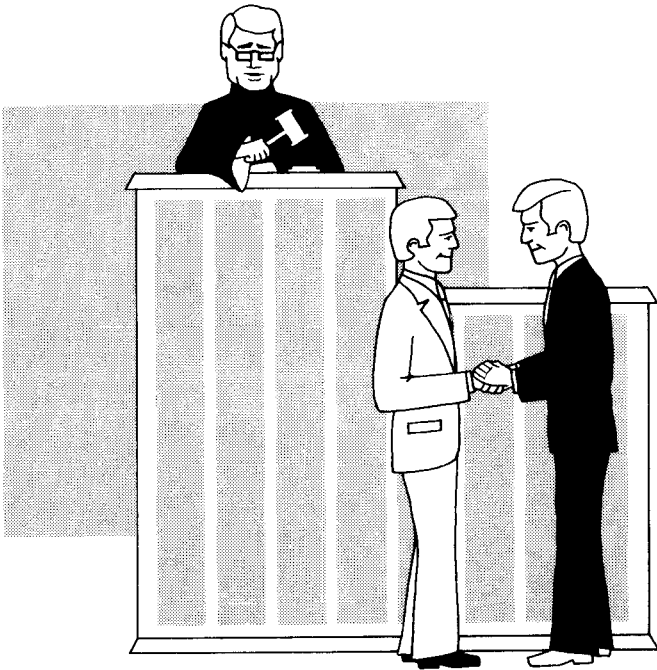
This manual is one of a series designed to introduce the software available with the SPERRY UNIVAC Operating System/3 (OS/3). The actual programming procedures required to use the software described are not included in this introductory series. Such detailed information is beyond the scope and intent of these manuals and is included in the appropriate User Guide and/or Programmer Reference.



data housekeeping

In conventional fiction, no one ever does routine chores. The television lawyer devotes his time only to the most sensational cases; the reporter is always involved in a big story; the doctor is constantly engaged in saving the lives of the most worthy patients.

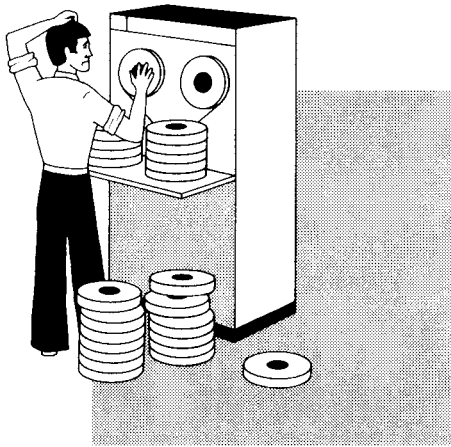
Unfortunately, life is quite different.



In fiction, we hardly ever see the commonplace — the dull, routine tasks that permeate nearly all activities. Long, tedious days of preparation by law clerks often go into a trial that requires only a few hours in court. The reporter usually covers Sunday socials or chases story leads that often lead nowhere. The doctor spends most of his day treating cuts, bruises, aches, and pains.



Just as a housewife spends much of her time doing housekeeping, so does a data processing installation devote much of its time to housekeeping. Many of their tasks have the attribute of drudgery in common.



Housekeeping done by a data processing installation consists not of domestic chores but of other jobs equally onerous. Some files must be duplicated while others must be reblocked and transferred to other media for long term storage. Other files simply need to be printed.

Programming for these tasks could be done by an installation's own programming staff. But, because of the sporadic nature of the runs and the fact that they are "one shot" operations, program modifications or reprogramming would be frequently needed. The resulting assembly or compilation time and test time would use too much of the installation's computer time, diverting busy personnel from more challenging tasks.

The solution to many of these user problems is provided by the UNIVAC Operating System/3 (OS/3) Data Utilities package. By punching a few control cards, you can perform many of the file modifications needed at most installations.

The UNIVAC OS/3 Data Utilities gain versatility through their handling of a wide variety of data storage media. This versatility allows them to transfer data from:

- card to tape
- card to disc
- tape to card
- tape to disc

- disc to card
- disc to tape
- card to printer
- tape to printer
- disc to printer

DATA PROGRAM

The capacity of Data Utilities to handle mixed media and to alter the contents of a file with its DATA program allows it to modify existing files for various purposes. It can easily insert and delete records, simplifying the modification of live files for special purposes such as creation of program test data. This can be carried a step further by the program's ability to reformat records, thus adapting files for input to programs for which they were not originally designed.

In keeping with the normal requirements of data processing installations, data files can also be transcribed onto the same medium. These operations include:

- card to card

— tape to tape

— disc to disc

The ability to edit data by adding, deleting, and reformatting records adds considerable power to the copy function of Data Utilities.

The uncertainty that sometimes exists after a file is duplicated as to the accuracy of the transcription is put to rest by the compare option of Data Utilities. You are allowed considerable flexibility in deciding just how accurate the match must be. And it can handle files on different media as well as the same medium, making it an ideal tool for checking the output of programs used for converting a tape-oriented processing system to a disc-oriented system, for example.

GANGPUNCH/REPRODUCE PROGRAM

A gangpunch/reproduce utility program is also available. With gang punching, you can punch a data field from a master card onto an output file. With the reproducing function, you can reproduce all or selected fields of a record onto an output card, relocate fields in the output records, or reproduce the original format. Program operation is initiated and controlled when the program reads special cards.

JPROCS

The UNIVAC OS/3 Data Utilities include a set of job control procedures called jprocs. By calling a jproc, you eliminate the need to repeatedly code identical sequences of job control statements required to execute a Data Utilities program. A single jproc call statement could generate most job control statements needed for your job, thus saving considerable keypunching time and reducing your job control programming effort.

reproducing and manipulating data files

The DATA program included with Data Utilities facilitates the reproduction and maintenance of your data files. With this program, you can quickly and easily perform a variety of operations that, otherwise, would require time consuming procedures. For example, you can:

- compare entire files or only areas of a file you select;
- delete or insert records in a file;

- rearrange existing records in a file;
- copy files from one storage device to another;
- print selected files; or
- work with nonindexed files having multiple positions.

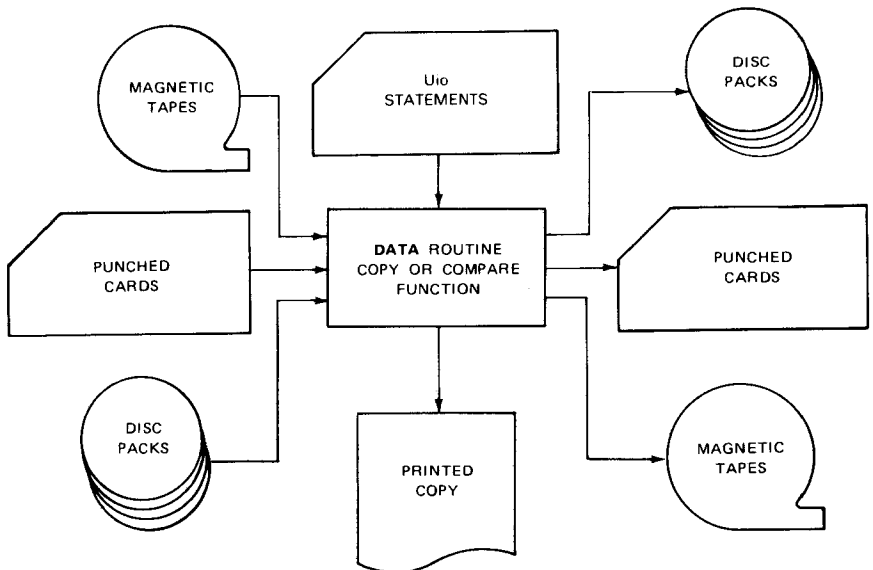
With these capabilities, here are some examples of possible uses for the DATA program:

- Data can be transferred from cards to disc or tape for long term storage with rapid access when needed.
- Records can be reformatted so that data in one program can be transcribed to a storage medium compatible with a different program.
- Specific areas of a file can be selected for testing or preparing reports, or the areas can be deleted.
- Contents of a file may be visually examined by reproducing them on printed copy.

- You can compare an input file with an output file to ascertain that no errors have occurred when copying files.

Uio STATEMENTS

The Data Utilities tasks are performed by using any of a number of available utility input/output (Uio) statements in your job control stream. The statements you supply contain all the information needed to perform the task.



The design places a high priority on ease of use. Ease of use is of special importance in a utility program as it is frequently used under circumstances that allow little time for preparation. The straightforward simplicity of the Uio statement approach requires little training and results in a much easier to use program.

Specific functions of the DATA program are easily implemented by including the appropriate Uio statement in your control stream. Each Uio statement basically consists of a mnemonic code that identifies the devices you require. The Uio codes are:

- UCC — card reader to card punch
- UCD — card reader to disc
- UCT — card reader to tape
- UCP — card reader to printer
- UDC — disc to card punch
- UDD — disc to disc
- UDT — disc to tape
- UDP — disc to printer

- UTC — tape to card punch
- UTD — tape to disc
- UTT — tape to tape
- UTP — tape to printer

Besides these codes, there are up to 26 different formats available for describing the types of files and devices that can be used. These are input and output parameters that designate special functions. Those most commonly used are default options so you do not need to code those parameters; the default value is automatically assigned to your job.

UTILITY MODIFIERS

A number of utility modifier control statements are also available that allow you to process certain fields that you select or to modify your files. These modifiers are used with the Uio statements for greater versatility in the DATA program. With utility modifier control statements you can:

- rearrange fields during a copy operation with the field select (FS) statement;

- select (SEL) or delete (DEL) specific fields by a search argument;
- request printing of page headings (HN) on the printed output;
- replace, delete, or insert records for corrections (COR); or
- handle nonindexed partitioned (PART) data files.

FILE CHARACTERISTICS

Uio statements may also be used to describe the input and output file characteristics. In addition to fixed-length blocks with fixed-length records, fixed-length blocks with variable-length records having key-field length specifications and data-field length specifications can be defined. Files with undefined block lengths can also be handled.

For magnetic tape files, you may specify both input and output tape label configurations; tapes with standard or nonstandard labels, or with no labels. Leading tape marks on files with nonstandard or no labels can be written on or omitted from the output tape file.

Block numbers on the input tape can be checked and written on the output tape or not; it's up to you. Input sequence numbers can be checked and output sequence numbers generated at the option of the user. On ASCII-coded input, fixed-, undefined-, or variable-length records can be processed to produce ASCII fixed-, undefined-, or variable-length output records. For card files, either EBCDIC or column binary coding can be used as both input and output. Stacker selection can also be specified.

USING A Uio STATEMENT

A great deal of effort was devoted to simplifying utility input/output operation. For example, in a card-to-disc operation, suppose you wish to copy employee records to a disc file, write them to the printer with headings on each page, rearrange certain fields of the input file, and delete certain records. The Uio statement you would code would appear as follows:

```
UCD DP, MY, PY, S2
```

UCD designates card-to-disc operation. The input and output block lengths default to the values of 80-byte record and block lengths because the designating A and B keyword parameters were

omitted. The print line size of 120 positions is also implied by the omission of the B keyword.

Keyword parameter DP specifies that output records are to be printed. The SAM format for input and output files will be used because other options were not specified.

Keyword parameters MY, PY, and S2 indicate that printer mismatches are to be ignored, page numbers and the date are to be printed on the first line of each page, and the printed output file is to be double-spaced. Omission of the print format keyword results in list format printing.

This Uio card is then followed by a group of six cards which specify the locations for moving specific card file columns, and deletion of certain records. Next, you would use the HN statement cards to describe the headings you want printed at the top of each page. These cards are followed by your data cards and the required job control statements to complete the job stream. A printed output shows that all your DATA program specifications have been executed successfully.

A variety of printer options are available, so that:

- Up to three title lines can be printed on each page.

- Records of any length can be specified.
- The printer can be used as a dual output device in conjunction with card, tape, or disc output.
- Printer mismatches can be ignored or recognized.
- Printing can be in EBCDIC, hexadecimal, or both codes.
- Page numbers and date can be printed or omitted.
- Printer spacing can be:
 - single space;
 - double space;
 - triple space; or
 - controlled by the first character of each record.
- Printing can be in "display" or "list" formats, for example:

SAMPLE PRINT - DISPLAY FORMAT - CHARACTER MODE

BLK#	REC#	BKSZ	RCSZ1080
1	1	80	80	012345678	UVWXYZ09*<
2	1	80	80	123456789	VWXYZ09*<0
:	:	:	:	:	:

SAMPLE PRINT - DISPLAY FORMAT - HEX MODE

BLK#	REC#	BKSZ	RCSZ53050
1	1	80	80	F0F1F2F3F4	D7D8D9E2E3	F5F6F7F8F9
				C1C2C3C4C5	E97C6C5C4C	
2	1	80	80	F1F2F3F4F5	D8D9E2E3E4	F6F7F8F9C1
				C2C3C4C5C6	7C6C5C4CFD	
:	:	:	:	:	:	:

SAMPLE PRINT - LIST FORMAT - CHARACTER MODE

0		80
0123456789ABCD	UVWXYZ0%*<	
123456789ABCD	VWXYZ0%*<0	
:	:	:

SAMPLE PRINT - DISPLAY FORMAT - BOTH MODE

BLK#	REC#	BKSZ	RCSZ1080
1	1	80	80	0123456789	UVWXYZ0%*<
				FFFFFFFFFFFF	EEEEEE7654
				0123456789	456789CCCC
2	1	80	80	123456789	VWXYZ0%*<0
				FFFFFFFFF0	EEEEEE7654F
				123456789	56789CCCC0
:	:	:	:	:	:

You can also go through the extensive list of options available through the Utility Modifier Statements and tailor the package more exactly to the needs of your application and the medium you are using. For every previously mentioned default choice, there are several others you could select.

FILE METHODS

Disc files organized according to the three most common access methods can be used as input or generated as output. The access methods allowed include:

- Sequential Access Method (SAM)
- Direct Access Method (DAM)
- Index Sequential Access Method (ISAM)

Fixed-length records with keys and fixed-length blocks with keys can be used as both input and output.

You are not limited to any one disc type. You can specify any of the discs offered with systems using OS/3, and you can even specify different types of discs for input and output.

DUPLICATING RECORDS

The basic operation of the Copy Function in the DATA program is to produce an exact duplicate of the input record on a selected medium. This operation is enhanced greatly by several options:

- You can specify the number of records or blocks that are to be processed.
- You can specify the logical record with which processing is to begin.
- Input can be column binary, EBCDIC (Extended Binary Coded Decimal Interchange Code), or ASCII (American Standard Code for Information Interchange — a code commonly used in a communications environment).
- Input records can be either in fixed or variable format within certain limitations.
- Input block numbers can be checked and output block numbers generated.
- Input sequence numbers can be checked and output sequence numbers generated.

- Output of a run can be duplicated on either the printer or card punch in addition to a specified output device.

SAMPLE DATA PROGRAM JOB RUN

Let's assume that your UNIVAC 90/30 Data Processing System has been installed and you wish to convert your files from punched card to disc. The following job control stream illustrates how simple a file conversion run could be, using the DATA program.

	1	10	20	30
1.	// JOB TESTFIL,1,A000			
2.	// DVC 20		// LFD PRNTR	
3.	// DVC 30		// LFD INPUT1	
4.	// DVC 51		// VOL D\$PO12	
5.	// EXT \$Q,C,,CYL,8			
6.	// LBL WRKFIL3		// LFD OUTPUT1	
7.	// EXEC DATA			
8.	/\$			
9.	UCD			
10.	/*			
11.	/R			
12.	// FIN			
13.	DATA CARDS			
14.	/*			
15.	/R			
16.	// FIN			

Line 1 shows the job beginning, jobname TEST-FIL, and the main storage requirements.

Line 2 assigns the printer to the job.

Line 3 assigns a card reader to the job to read the input file.

Line 4 assigns a disc file unit to the job to store the output file on disc volume DSP012.

Line 5 obtains disc space, specifies sequential (SAM) format, and allocates contiguous cylinder space on eight cylinders.

Line 6 identifies the output file by file name.

Line 7 requests loading and execution of the DATA program.

Line 8 designates the start-of-data for the DATA program.

Line 9 is the Uio statement that identifies the job as a card-to-disc operation of the DATA program.

Line 10 designates the end-of-data to the DATA program.

Line 11 indicates the end-of-job.

Line 12 terminates the card reader.

Line 13 indicates where the input file data cards are placed in the control stream.

Lines 14 and 15 designate the end-of-data and end of your job.

Line 16 terminates the card reader.

Because all parameters that form the operand field of the UCD statement have been omitted, all the default options are assumed by the data utility program. Thus, it is assumed that:

- the input file contains fixed-length records with record and block lengths of 80 characters each;
- the output file will contain fixed-length records with record and block lengths of 80 characters each;
- the discs used are from the UNIVAC 8416 Disc Subsystem;
- the input is in EBCDIC code;

- the input punched card file is to be copied onto disc; and
- the disc check operation will not be performed.

gang punching cards and reproducing data

The Gangpunch/Reproduce Program can not only perform all of the standard gang punching and reproducing operations, such as:

- straight reproducing;
- selective reproducing;
- gang punching; and
- combined gang punching and reproducing;

but it can also perform additional operations, such as:

- rearranging existing data fields;
- checking the sequence numbering of a card deck for ascending or descending sequence;

- punching constant data onto output cards;
- matching cards;
- controlling major-minor punching; and
- selecting an output stacker on the card punch.

The Gangpunch/Reproduce Program offers two advantages. First, as previously indicated, is the ability to perform operations not normally available on a reproducing punch unit. Second is the reduction in setup time; instead of setting up the operation by wiring the unit's control panel, the operation is initiated and controlled by specification cards read by the program. Frequently-used operations can be stored on specification cards rather than on wired control panels, thereby achieving significant cost reductions.

GANGPUNCH/REPRODUCE JOB STREAM

The Gangpunch/Reproduce Program uses a job stream divided into three parts to simplify programming. These are:

- the job control cards that assign files and devices to accommodate your job;

- the gangpunch/reproduce control cards that:
 - check the sequence numbering (CTL) of a data deck;
 - select (SEL) detail or master cards in the runstream; and
 - identify input and output data field (FLD) locations and request punching of sequence numbers in ascending order or gang punching of constant data.

- data cards that contain your input information (these may be master or detail cards or combinations of them).

SAMPLE GANGPUNCH/REPRODUCE PROGRAM

Control cards in the Gangpunch/Reproduce Program can manipulate the contents of the data deck in either one- or two-step operations. The following example uses a single job step to check the sequence numbering in a data deck containing no card type designation.

	1	LABEL	10	OPERATION	20
JOB CONTROL CARDS	1.	// JOB GANGPCH	,3000		
	2.	// DVC 20		// LFD PRNTR	
	3.	// DVC 30		// LFD INPUT1	
	4.	// DVC 40		// LFD OUTPUT1	
	5.	// EXEC GANGPCH			
	6.	/R			
	7.	// FIN			
GANGPUNCH REPRODUCE CONTROL CARDS	8.	SELA 011C1			
	9.	SELB 011C1			
	10.	CTL	11004	C	
	11.	FLD A0105CA01			
	12.	FLD B0105CB01			
DATA DECK	13.	15739	8956		
	14.	32567	8959		
	15.	32569	9067		
	16.	78432	9058		
	17.	95731	9070		
	18.	/*			
OUTPUT DECK	19.	15739	8956		
	20.	32567	8959		
	21.	32569	9067		
	22.	95731	9070		
	23.	/*			

Line 1 specifies the job name GANGPCH with a minimum storage requirement of 3000₁₆ bytes.

Line 2 assigns the desired printer for printing error messages and the contents of each output card.

Lines 3 and 4 define the card reader file as INPUT1 and the card punch file as OUTPUT1.

Line 5 initiates the gangpunch/reproduce program.

Lines 6 and 7 terminate the job control stream.

Line 8 specifies that all cards having a character equal to 1 in column 1 are A-type detail cards.

Line 9 specifies that all cards not equal to 1 in column 1 are B-type detail cards (lines 8 and 9 ensure that all data cards are checked).

Line 10 is the control card that requests sequence checking for ascending order beginning in column 10 for a length of four bytes.

Lines 11 and 12 are field definition cards that define the type of input card, location and length of input data, and what portion of the input character to reproduce.

Lines 13 through 17 are data cards which are read and punched to produce the output deck. Each card also supplies a printed output.

Line 18 specifies the end-of-data of the Gangpunch/Reproduce Program.

Lines 19 through 22 are the output deck.

With this example program, card 16 in the data deck is found to be out of sequence. No output card is punched for that line and the following message appears in the listing:

```
PREVIOUS CARD OUT OF SEQUENCE*  
CARD NOT PUNCHED*
```

data utility jprocs

To further simplify the use of Data Utilities, job control procedure (jproc) statements can be used when running a disc-to-disc (UDD), disc-to-tape (UDT), or tape-to-disc (UTD) operation. The jproc call statements can also be used when comparing two files.

Jprocs eliminate the need to repeatedly code identical sequences of job control statements. The jprocs are stored in the job control stream library file, and are used by placing the jproc call statement in your job control stream at the same point you would have otherwise placed your job control statements. All the proper sequences of job control statements are then generated when you run the job.

Input and output files can be accessed in a variety of combinations, and may be copied or relocated to the same volume or a different volume when using jproc call statements. Input and output file access methods that may be used for copy or compare operations are:

- ISAM fixed or variable disc files;
- SAM fixed or variable disc files;
- DAM fixed disc files; and
- SAM fixed or variable tape files.

All of the above-listed access methods may be used for input and combined in a variety of similar methods for output files for copy or compare operations. Compatibility between disc and tape unit numbers is assured by special jproc call statements which assign logical unit numbers to disc and tape units.

EXAMPLE OF JPROC USE

This example illustrates a SAM disc-to-SAM disc copy operation using the UDD jproc call state-

ment for a simple disc-to-disc copy. It is assumed that space has been previously allocated for the output file. In this example, the input and output files are on the same disc.

LABEL	OPERATION	OPERAND	5	COMMENTS
1	10	20	30	40 50
1	//	JOB DCDCCPY1		
2	//	UDD IN=(DSP001,LABEL1),OUT=(DSP001,LABEL2,,INIT)		
3	//	*		
4	//	UDD		
5	//	*		
6	//	*		
7	//	FIN		

Line 1 indicates the job name.

Line 2 indicates the jproc being called is UDD. The *IN* parameter indicates the file identifier (LABEL1) of the input file and the file and serial number (DSP001) of the disc where the file is located. The *OUT* keyword parameter indicates the file identifier (LABEL2) of the output file and the volume serial number (DSP001) of the disc where the file is located.

Since the file is assumed to be previously allocated, *INIT* is used to indicate that any information that may be on the file is to be overwritten by the new information.

Line 3 indicates the start of data.

Line 4 indicates the Data Utility control statement. The input and output file type are both defaulted to SAM file, and the default block size and record size for input and output are 80.

Lines 5, 6, and 7 are standard end-of-data and end-of-job control statements.

Now let us examine the expanded job control stream generated by the example.

LABEL	OPERATION	OPERAND
1	10	20
1.	//	JOB DC.DCCPY1
2A	//	DVC 20
2B	//	LFD PRNTR
2C	//	DVC 50
2D	//	VOL DSPool
2E	//	LBL LABEL1
2F	//	LFD INPUT1
2G	//	DVC 50
2H	//	VOL DSPool
2I	//	LBL LABEL2
2J	//	LFD OUTPUT1, INIT
2K	//	EXEC DATA
3.	/	\$
4.		UDD
5.	/	*
6.	/	&
7.	//	FIN

Obviously, the jproc call statement used in the first example has reduced your programming time by eliminating the need to generate nine extra job control stream cards.

summary

The transformation of raw data into useful information is the function of all data processing equipment and the goal of all users. The UNIVAC Operating System/3 (OS/3) has been designed to accomplish this transformation of bits and bytes into facts and figures in the most efficient manner possible. This efficiency is made possible by the careful mating of the software and hardware components in the system, a design philosophy respected throughout the development of OS/3. As one of these software components, the UNIVAC OS/3 Data Utilities contribute greatly to the efficiency of the overall task of transforming data into information by freeing data processing personnel from some of the routine chores and allowing them to concentrate on more creative activities.

Please note that a more detailed description of the features and capabilities of UNIVAC OS/3 Data Utilities, complete with illustrations and examples drawn from actual practice is available in the *UNIVAC Operating System/3 Data Utilities User Guide/Programmer Reference*.

