

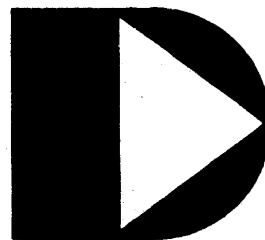
DATASHARE PROGRAM GENERATION SYSTEM DSGEN User's Guide

Version 1

October 20, 1976

Model Code No. 50246

DATAPPOINT CORPORATION



The leader in dispersed data processing™

DATASHARE PROGRAM GENERATION SYSTEM
DSGEN

User's Guide

Version 1

October, 1976

Model Code No. 50246

PREFACE

DSGEN is a series of DATABUS programs which provide for efficient forms generation and data entry, printing, and updating capability for a DATASHARE 3 or DATABUS 11 system.

NOTICE

Datapoint strongly recommends that its customers use Datapoint Customer supplies. These disks, diskettes, cassettes and ribbons are certified by Datapoint to meet all Datapoint hardware specifications for consistent optimum performance.

TABLE OF CONTENTS

	page
1. INTRODUCTION.	1-1
2. SYSTEM GENERATION	2-1
2.1 Loading From Cassette	2-1
2.2 Compiling the DSGEN System Programs	2-2
2.3 Required Common Data Area	2-2
2.4 Necessary Programs	2-3
3. SYSTEM OPERATION	3-1
4. ENTRY PROGRAM GENERATOR - DSENTRY	4-1
4.1 Input to DSENTRY	4-1
4.2 Output of DSENTRY	4-1
4.3 Creating a Program Source File	4-2
4.3.1 ISAM or Sequential Output Files	4-2
4.3.2 Re-starting from a Prior Screen	4-3
4.4 Creating the Screen	4-4
4.4.1 Entering the Screen	4-5
4.5 User Comments	4-6
4.6 Field Edits	4-7
4.6.1 Field Edit Classifications	4-7
4.6.2 Entering Field Edits	4-8
4.6.3 Clearing Field Edits	4-9
4.6.4 Level 2 Edit Definitions	4-9
4.6.4.1 <u>Type 1 (R)equired Fields</u>	4-9
4.6.4.2 <u>Type 2 Field (T)ypes</u>	4-9
4.6.4.3 <u>Type 3 Field (P)rograms</u>	4-10
4.6.5 Field Edit Requirements	4-10
4.6.5.1 Numeric field format	4-10
4.6.5.2 Auxillary ISAM verify	4-11
4.6.5.3 Table-Lookup	4-11
4.6.5.4 Range Check	4-11
4.7 Screen-Image Printout	4-11
4.8 Resequencing Fields	4-12
4.8.1 How to re-sequence fields.	4-12
4.8.2 Example of Re-Sequencing	4-13
4.9 DSENTRY Program Phases	4-14
4.10 DSENTRY Execution-Time Messages	4-14
4.10.1 Phase DSENTRY Messages	4-14
4.10.2 Phase DSENTRY1 Messages	4-15
4.10.3 Phase DSENTRYC Messages	4-16
4.10.4 Phase DSENTRYE Messages	4-16
4.10.5 Phase DSENTRYP Messages	4-17

4.10.6	Phase DSENTRY5 Messages	4-17
4.10.7	Phase DENTRY2 Messages	4-18
4.10.8	Phase DENTRY3 Messages	4-19
4.10.9	Phase DENTRYD Messages	4-19
4.10.10	Phase DENTRYN Messages	4-19
4.10.11	Phase DENTRYT Messages	4-19
4.10.12	Phase DENTRYR Messages	4-20
4.10.13	Phase DENTRY4 Messages	4-20
4.10.14	Phase DENTRY5 Messages	4-21
4.10.15	Phase DENTRY6 Messages	4-21
5.	PRINT PROGRAM GENERATOR - DSPRINT	5-1
5.1	Input to DSPRINT	5-1
5.2	Output of DSPRINT	5-2
5.3	Creating a Program Source File	5-2
5.3.1	ISAM or Sequential Input Files	5-2
5.4	Creating the Print Program	5-3
5.4.1	Field Elimination	5-3
5.4.2	Re-Instating an Eliminated Field	5-4
5.4.3	User Comments	5-5
5.4.4	Control Groups (Level Breaks)	5-5
5.4.4.1	Defining Level Breaks	5-5
5.4.4.2	Accumulating Sub-Totals	5-6
5.4.5	Sequence of Fields in Printout	5-6
5.4.6	Report Headings	5-6
5.4.6.1	Page Heading	5-6
5.4.6.2	Field Heading	5-7
6.	UPDATE PROGRAM GENERATOR - DSUPDATE	6-1
6.1	Input to DSUPDATE	6-1
6.2	Output of DSUPDATE	6-2
6.3	Creating a Program Source File	6-2
6.4	ISAM or Sequential Input Files	6-2
6.5	User Comments	6-3
6.6	Field Edits	6-3
6.6.1	Auxillary ISAM Verify	6-3
6.6.2	Table-Lookup	6-4
6.6.3	Range Check	6-4
6.6.4	Data Entry Sequence	6-4
6.7	DSUPDATE Program Phases	6-4
Appendix A.	SAMPLE DSGEN SCREEN LAYOUT	A-1
Appendix B.	SAMPLE DSPRINT REPORT (DETAIL LIST)	B-1
Appendix C.	SAMPLE DSPRINT REPORT (ONE LEVEL BREAK)	C-1
Appendix D.	DSPRINT GENERATED PROGRAM LOGIC FLOW	D-1

Appendix E. SCREEN-IMAGE WORK FILE DEFINITION

E-1

INDEX

CHAPTER 1. INTRODUCTION.

Using the DSGEN system, a data-entry screen is designed by a programmer and a DATABUS program is generated from the screen image. Optionally, the programmer may include field edit definitions, define range-checks, table-lookups, etc. After the form is entered, the data-entry sequence can be modified so the entry operator can input the fields in any order.

The screen-image file created by program DSENTRY is then used by the other programs of the DSGEN system to create a print-out of the database and update the database.

The programs included in the DSGEN System are:

DSGEN - Master Menu Program (Selects one of the following programs)
DSENTRY - Entry Program Generator
DSPRINT - Print Program Generator
DSUPDATE - Update Program Generator

The output of DSGEN is DATABUS source files that can be directly compiled with the DOS. DATABUS compiler, or, using the DOS. EDIT command, additional complex editing can be inserted by the programmer. The screen layout is saved as a text file in case additional programs are to be created.

DSGEN programs will execute on any DATASHARE port configured for at least 670 bytes.

CHAPTER 2. SYSTEM GENERATION

2.1 Loading From Cassette

Normally, the DSGEN System is supplied as DATABUS Object Code files contained on a cassette tape. The cassette is in the DMF (DOS Multiple File) format which includes a directory of the files on the tape. All that is necessary to load the system files to disk is to have the MIN program catalogued on the system and to give the DOS command:

```
MIN ;A
```

The MIN (Multiple IN) program will be activated and will display the date of creation of the tape, the file names in the tape directory, and each file name as the file is being loaded. If the file already exists on the disk, the MIN program will ask if it is to be overstored. The operator can decide to overstore the file or can tell it not to overstore the file, in which case MIN will allow the file to be stored under a different name. Consult the DOS USER'S GUIDE for further information on this procedure.

After the DSGEN programs have been loaded onto your system, it will be necessary for you to run the DOS. FILES program to make sure the programs just loaded were not 'fragmented' by the MIN program. Make sure that none of the programs occupy more than one (1) "Segment".

NOTE: Current versions of DATASHARE 3 and DATABUS 11 Interpreters will not execute any programs that span more than one segment. The interpreters which have this restriction are:

- DB11 2.1
- DS35500 1.3
- DS3A3360 1.2
- DS3B3360 1.2
- DS3A3600 1.2
- DS3B3600 1.2
- PSDS3 1.1

To run the FILES program, give the DOS. command:

FILES DS/DBC;D

A typical "FILES" list follows. Note that none of the /DBC files occupy more than one sector.

PROGRAM PHASE	# of SEGMENTS					
-----	-----					
DSENTRY /DBC (016)	SEG	1	CYL	51	CLUSTER 6	SIZE 54
DSENTRY3/DBC (124)	SEG	1	CYL	103	CLUSTER 0	SIZE 60
DSGEN /DBC (047)	SEG	1	CYL	9	CLUSTER 1	SIZE 12
DSPRINT /DBC (050)	SEG	1	CYL	88	CLUSTER 6	SIZE 66
DSUPDAT1/DBC (106)	SEG	1	CYL	93	CLUSTER 5	SIZE 24
DSUPDATE/DBC (054)	SEG	1	CYL	9	CLUSTER 4	SIZE 66

If any of the program phases take more than one segment, you will have to free up some disk space and re-load that phase (using MIN).

2.2 Compiling the DSGEN System Programs

If you have a licensed copy of the DSGEN System Program Source Files, you can MIN them to your system as described above. Text files can span many segments without any problems. After the MIN program has run to completion, use the DOS CHAIN command to compile DSGEN as follows:

```
CHAIN DSGEN/CHN;<parameters>
```

Use the 'LIST' command for a listing on your system printer, or, use the 'SPOOL' command if you want the printed output of the compilation to be in print-image files. The 'LIST' and 'SPOOL' options require the 'DATE' parameter. If listings are not required, use the 'NO' parameter.

2.3 Required Common Data Area

DSGEN expects Port Number in the first two bytes of common, and the date in the next 8 bytes (MM/DD/YY) format. The correct format for port number and date is in the file 'COMMON/TXT'. The source programs generated by DSGEN will also automatically generate code to include 'COMMON/TXT' in generated source programs.

2.4 Necessary Programs

Before the DSGEN system can be executed, your system must have either a DATASHARE 3 interpreter (DS3/CMD) or the DATABUS 11 interpreter (DB11/CMD). Also, a DATABUS MASTER program must exist that includes the correct common data area definitions.

CHAPTER 3. SYSTEM OPERATION

To execute DSGEN, enter 'DSGEN' once DATASHARE is running. Alternately, enter the DOS command: DB11 DSGEN.

NOTE: The messages generated by DSGEN are in a unique format. The prompt contains the expected response within parentheses (). An example of a prompt that asks a YES/NO type question:

(Y)es or (N)o ?

DSGEN will expect the letter embedded in the parentheses as a response.

CHAPTER 4. ENTRY PROGRAM GENERATOR - DSENTRY

Many data-entry programs are simply 'iterative' loops, i.e.,

1. Display Screen;
2. Keyin Variables;
3. Edit Variables;
4. Write Variables to Data Base;

The programmer's task is similar, no matter which data entry task is assigned to him. DSENTRY will allow the programmer to easily create a functional DATABUS program, since all that is necessary to accomplish program generation with DSENTRY is defining a screen format.

Simple data-entry programs can be created in a matter of minutes; complex programs take somewhat longer. Since the output of DSENTRY is a standard DATABUS source file, any changes that the programmer wishes to make to the program generated by DSENTRY can be done using the DOS. EDIT command.

4.1 Input to DSENTRY

The input to DSENTRY is a programmer, sitting at a DATASHARE Terminal. DSENTRY will interact with the programmer on a line by line basis (inputting the screen) or on a field by field basis (comments, edits).

4.2 Output of DSENTRY

DSENTRY will automatically create source code for:

1. ISAM file opening sequence;
2. Sequential file opening sequence;
3. Logic to Extend an existing sequential output file;
4. Open auxillary ISAM "lookup" files;
5. Automatic field back-tabbing (using the '<' character);
6. Automatic field duplication (using the '>' character);
7. In-line date verification;
8. In-line range-checking of variables;

9. In-line table-lookup validation;
10. Reserved (display-only) variables;
11. File-closing sequence;
12. End of job sequence.

4.3 Creating a Program Source File

The program that will be created by DSENTRY is a standard 'text' file. You may edit the file, compile it, list it, etc.

The program generated by DSENTRY needs a name. Program names must conform to the DOS standard naming conventions, i.e., up to 8 letters and numbers with no special characters or spaces embedded.

4.3.1 ISAM or Sequential Output Files

The output disk file that your DSENTRY generated program will create can either be an ISAM file, or a sequential file. ISAM files are created by executing the INDEX utility program. Direct files (including sequential access) are standard Datapoint space-compressed files. All of the variables included in a screen layout (with the exception of "Display-Only" variables) will be written to a single record in the order in which they appear on the screen.

DSENTRY will now ask:

OUTPUT FILE TYPE:(S)EQUENTIAL OR (I)SAM ?

Respond with the letter 'S' or 'I' as necessary. If you aren't sure which file type, consult the DOS DATABUS (DBCMP) User's Guide for a discussion of the different file structures.

If you respond 'Isam', you will later be asked the following question about the Prime ISAM key:

PRIME ISAM-KEY IS ON LINE-LETTER (?) WHICH FIELD (?)

The variable at that location will be used to insert the records into your output file. In order to prevent problems caused by Duplicate Keys, you can specify that the same field as the Prime Key is also to be used in an auxillary ISAM lookup (See

section 0.0.0). If the program gets a 'Hit' on the auxillary file, the operator can be notified that this is a bad condition.

4.3.2 Re-starting from a Prior Screen

After a program has been created using DSENTRY, it is often useful to involve the actual user of the program in a critique of the screen layout. These criticisms often make it necessary to change the screen layout. When conventional coding techniques are used, this is often difficult or impossible. With DSENTRY, however, it is an asy matter to modify the screen to suit the user.

DSENTRY creates two work files used for generating a program. The most important work file is the 'SCREEN' file. This file actually contains the screen layout in mirror-image format. When the programmer wants to change the screen layout, he is simply changing the screen file. After the initial screen (and test program) have been created, the screen file exists as a text file on disk. It has a file specification of:

SCREENq/TXT

Where 'q' represents a port-letter (port 1=A, port 2=B, etc).

If a screen is to be used for later modifications, simply rename the screen file (using the DOS NAME command) when you are through with the first pass through DSENTRY. Later, you can re-start from that base by answering the question:

DO YOU WANT TO RE-START FROM A PRIOR SCREEN ?

If a re-start is to be done, respond Yes, else, respond No. If you respond Yes, you will be asked:

WHAT IS THE NAME OF THE OLD SCREEN FILE ?

Simply enter the name of the saved screen file. If you enter a name that does not exist, or if the screen file is not of the proper format, DSENTRY will simply create a default screen file (SCREENQ/TXT). If the file is of valid format, DSENTRY will display the message 'SUCCESSFUL RESTART'.

NOTE: the screen file is not 'transportable', except by the use of the DOS. COPY command. You cannot expect to use the DOS. MOUT command to save the screen on cassette. This feature will provide the user with some security for his valuable screen layouts.

4.4 Creating the Screen

There are two kinds of data that DSENTRY is concerned with:

1. Variables;
2. Constants (called operator prompts)

Any character can be used in a constant except the UNDERSCORE () character.

Any character can be used in a variable except the SLASH (/) character.

Anywhere the UNDERSCORE () character is placed on the screen, a variable is automatically defined.

Embedding the SLASH (/) character in a series of UNDERSCORE characters implies a DATE field. If the character string of Underscore and Slash characters is not exactly / / DSENTRY will tell you that there is an error on that line. Don't forget to further define the Date field using the correct Field Edit code.

NOTE: DATABUS 'forcing character' (the # character) can be used. Also, the double-quote (") can be used. Wherever these two characters are used on the screen, DSENTRY will generate the correct code in the source program by preceding the character with a forcing character.

Example of an operator prompt and a variable:

CUSTOMER NAME: _____

The data-entry operator will enter up to 20 characters where the underscore characters are.

At this point, Numeric Variables are no different from

character variables. A numeric field that has a format of 5.2 would be entered by the programmer as eight ' ' (_____). This reserves room for the decimal point - THE PROGRAMMER DOESN'T ENTER THE DECIMAL POINT YET!

Dates are often used in data-entry programs, and are a constant source of annoyance to the programmer who has to write the date edit routines. To define a standard date format to DSENTRY, the programmer enters the special character '/' the slash character in the following manner:

(Prompt)INVOICE DATE __/__/__

During the Field Edit phase, the operator can tell DSENTRY that a standard date check should be performed on the field. See chapter on Field Edits (section Field Type Definition).

A special case of a variable is "Display-Only". This variable type is generally used when the result of an auxiliary ISAM lookup is simply to be displayed on the screen, or when a special message to the operator is to be displayed. Display-only variables will be included in the generated source program as 'comment' lines. It is the responsibility of the programmer to include the appropriate logic for the use of the variable (using the DOS. EDIT command on the DSENTRY generated program).

4.4.1 Entering the Screen

At this point, you are ready to enter a screen layout. The letters A thru W will be displayed down the left side of the screen. Each letter represents one line. For instance, the letter 'C' represents line three (3). The bottom line of the screen will be used for further machine-programmer communications. Refer to Appendix A for a sample screen. DSENTRY will ask you the following question about the next line to be entered:

SAMPLE ENTRY: NAME _____ LINE-LETTER (A-W) ?

To enter a line on the screen, enter a valid letter (A thru W). DSENTRY will erase that line from the screen (and the screen-file), and wait for you to enter something. In the sample

screen, Line-A contains the screen title:

ACME ARMADILLO MEAT SPECIALTIES CORPORATION

There are no variables on the line, however, this makes the rest of the screen more presentable to the user. Using blank lines may also make the screen more presentable.

Line-B contains operator prompts and input variables for:

INVOICE # _____ INVOICE DATE __/__/__

The operator will actually enter data where the underscores are. Refer to the Field Edit Section for an explanation of the Invoice Date Field.

Note: the maximum number of variables on one line is 14.

When all of the necessary prompts and variables are defined the way you want, tap the Enter key. DSENTRY will now ask:

ARE YOU DONE (Y)ES OR (N)O ?

A response of 'Yes' will tell DSENTRY that no more lines are going to be entered.

4.5 User Comments

Now that the screen has been created, it is desirable to comment the source program that is being created. A single screen line can have as many as 14 fields on it, and there can be up to 23 lines of screen data. It would be undesirable to generate a program of such size without having some kind of comments associated with each of the fields. DSENTRY will elicit comments from the programmer for each field on the screen that has 1 or more variables on it. DSENTRY will display the message:

COMMENT FOR LINE (?) FIELD (?) (.....)

If a comment existed (due to a re-start), it will be displayed. The operator may elect to change it, or, by tapping the Enter key, leave it alone. The user comment area is 30 bytes long, however, the last 4 bytes are reserved for DSENTRY if the field type is numeric.

DSENTRY will 'blink' the first position of the field for which it is asking that a comment be entered. A handy technique is to tap the Enter key a couple of times - this will bring your eye right to the field in question. The user comment should be a quick note to the person who will 'read' the generated source program.

If a user comment already exists due to a restart, that comment will be displayed. If you want the same comment, just tap the Enter key. Otherwise, enter the new comment.

In the sample screen (Appendix A) the comment for line 'B', field '3' would be 'INVOICE NUMBER'.

4.6 Field Edits

Most data-entry programs require some type of field validation. For example, an employee number that has to be verified before payroll data can be entered might require an access to the master employee database. In the case of a date, correct month, day, and year must be checked.

4.6.1 Field Edit Classifications

DSENTRY allows for three major edit classifications:

1. Required fields;
2. Field Type definition;
3. Field Programs.

4.6.2 Entering Field Edits

After the user comments have been entered, DSENTRY interacts with the programmer in a manner that makes entering field edits quite simple. DSENTRY asks for field edits in a hierarchial manner:

LEVEL 1 (R)equired Edits	(F)ield Types	Field (P)rograms
LEVEL 2 (R)equired (F)ill Controlled (B)oth Req & Fill di(S)play-only	(N)umeric Type (D)ate Check s(H)ift-Invert	(I)SAM Verify (T)able-Lookup r(A)nge Check
LEVEL 3 Line Letter for Edit	Field Number on that Line	

The letter in parenthesis indicates the word that describes the edit.

Sample operator-machine dialog to request a numeric field edit on line 'G', field '3' (ZIP code in example):

Machine (Level 1):

(R)equired Fields Field (T)ypes Field (P)rograms (N)o More ?

Programmer.Response is 'T' (for Field Types)

Machine (Level 2):

Field Type: (N)umeric (D)ate s(H)ift-key inversion

Programmer.Response is 'N' (for Numeric Format)*

Machine (Level 3):

(N)umeric Field on Line-Letter ? Which Field (1- ?)

Programmer.Response is 'G' (Line-Letter) and '3' (Field #)

DSENTRY will now ask for additional (N)umeric fields. If there are no more numeric field edits, tap the Enter key, DSENTRY

will return to Level 2 questions, then to Level 1 questions (in that order). In the sample screen, you can tell DSENTRY that Line G, Field 6 is also Numeric, etc.

4.6.3 Clearing Field Edits

If the programmer is unsure of the particular edits on a field (because of a restart) or if he simply wants to clear all the edits on a field, respond with the letter 'C' to the Level 1 question. DSENTRY will immediately go to Level 3 (Line letter and field number).

4.6.4 Level 2 Edit Definitions

4.6.4.1 Type 1 (R)equired Fields

(R)equired - means that at least one character must be entered into the field.

(F)ill-controlled - means that every character of the field must be entered.

(B)oth Required and Fill-controlled - means that every character must be entered.

Di(S)play-only - means the programmer is just reserving room in the program for a variable, but the data-entry operator will not actually enter anything there. This is used when the programmer expects to modify the generated program.

4.6.4.2 Type 2 Field (T)ypes

(N)umeric - means only a valid Databus numeric variable can be entered.

(D)ate - means a standard date (MM/DD/YY) must be entered.

S(H)ift Inversion - means the field to be entered will be in

inverted (typewriter) format.

4.6.4.3 Type 3 Field (P)rograms

(I)SAM-verify - means an auxillary ISAM file is to be checked for the presence or absence of the variable. (See section 6.6.2).

(T)able-lookup - means that a table is to be searched for the presence or absence of the variable.

R(A)nge-check - means the variable is to be within a particular range.

4.6.5 Field Edit Requirements

If a field was defined as Numeric-type, or had a Field Program associated with it, additional programmer input is necessary. For example, a numeric field has a certain decimal format; an auxillary ISAM verify can be either good or bad, etc. As DENTRY is generating the mainline of the user program, it may ask for these responses. The field in question is 'blinked', and the line-letter and field number are displayed on the bottom line of the screen.

4.6.5.1 Numeric field format

If the programmer defined a field as Numeric (see section on field edits) DENTRY will now ask for the decimal format of the field. If this information is already available (restart), the operator can just tap the Enter key. The decimal format must 'fit' the form defined by the programmer. For example, a field that has 5 positions before and 2 positions after the decimal point should have 8 positions reserved on the screen. If so, simply enter '5.2' when DENTRY asks for the format.

NOTE: The Numeric Format is saved in bytes 27-30 of the User-Comment.

4.6.5.2 Auxillary ISAM verify

If the programmer requested that an ISAM verify was to be performed on a field, DSENTRY needs to know whether the resulting access will indicate a good condition or a bad condition. If a record must be present on a database, it is considered a good condition, if a record is absent from a database, it is considered a bad condition. The ISAM verify code generated by DSENTRY will just be a simple ISAM read into a 1 byte work area. If a real database is to be used for field validation, it is the programmer's responsibility to include the correct read statement in the generated source program.

4.6.5.3 Table-Lookup

If the programmer requested that a table-lookup be performed on a field, DSENTRY will ask for the number of entries in the table. For instance, a part number prefix might be 'AA' or 'BB' or 'CC' - if it isn't, the operator must re-key the field. On the other hand, entering a code of 'Y7' or 'W3' might be invalid, in which case, the operator must re-key the field. DSENTRY will need to know if the resulting lookup is good (meaning the data must be present in the table) or bad (meaning the data must not be present in the table). Now, DSENTRY will accept the actual table values from the programmer.

4.6.5.4 Range Check

If the programmer requested that a Range-Check be performed on a field, DSENTRY will ask for the lowest and highest values permitted. No attempt is made to see that the two values are arithmetically correct. It is entirely up to the programmer to assign good upper and lower limits. Range checks can be performed on alphameric or numeric fields.

4.7 Screen-Image Printout

If the programmer wishes, a printout of the exact screen image can be made after the field edit phase. Simply respond 'Y'es when DSENTRY asks if you want a printout of the screen. The resulting printout is useful as a work sheet, and can be included in the final documentation of the generated program.

4.8 Resequencing Fields

If the sequence of the fields as the programmer enters them is going to be different from the sequence that the data-entry operator is expected to enter, DSENTRY will allow you to alter the sequence of the variables. Refer to the sample screen in Appendix A. The programmer would enter the screen just as you see it, however, it is undesirable for the data-entry operator to enter the fields top-down and left-to-right. Logically, the data-entry operator will be expected to enter the 'SOLD-TO' information in toto before entering the 'SHIP-TO' information. The entry sequence of the rest of the fields is irrelevant in this discussion.

4.8.1 How to re-sequence fields.

If a Re-start was performed, and if no additional variables were added to the screen, DSENTRY will ask:

DO YOU WANT THE SAME SEQUENCE AS LAST TIME ?

If you respond 'Yes', DSENTRY will use the sequence information from the last time the screen was used to create the current program. If the sequence information was lost, or can't easily be ascertained, DSENTRY will ask:

DO YOU WANT TO RE-SEQUENCE THE DATA FIELDS ?

If you respond 'No', DSENTRY will default to a sequence of Top-Down and Left-to-Right. In other words, each line will be inspected (starting at Line-A) and all the fields on that line will be generated into the source program in the sequence that they appear. Then, the next line (LINE-B) will be inspected, etc.

If you respond 'Yes' to the preceding question, DSENTRY will ask the following question about the next field:

NEXT FIELD IS ON LINE-LETTER ? _ WHICH FIELD ___?

Reply with the Line Letter and the field number that is to be next in sequence (when the program is to be run). DSENTRY will

display an asterisk (*) in the first position of the field so you can keep track of the remaining fields. Any fields that you don't specifically supply sequencing information for will follow all fields that you do supply sequencing information for. The unspecified fields will follow the sequenced fields in the following sequence:

Top-down & Left-to-Right.

4.8.2 Example of Re-Sequencing

In the sample screen in Appendix A, the following Line-Letters and Field Numbers were entered:

Line B Field 1 Invoice #
Line B Field 2 Invoice Date
Line D Field 1 Sold-to Name
Line E Field 1 Sold-to Address 1
Line F Field 1 Sold-to Address 2
Line G Field 1 Sold-to City
Line G Field 2 Sold-to State
Line G Field 3 Sold-to ZIP code

Since the rest of the fields are already in the correct data-entry sequence, no additional sequencing information was given! DENTRY generated sequence information for the rest of the fields in the following sequence:

Line D Field 2 Ship-to Name
Line E Field 2 Ship-to Address 1
Line F Field 2 Ship-to Address 2
Line G Field 4 Ship-to City
Line G Field 5 Ship-to State
Line G Field 6 Ship-to ZIP code
Line J Field 1 Item number
Line J Field 2 Quantity
Line J Field 3 Description
Line K Field 1 Invoice Total

CAUTION: If you hit the INTERRUPT KEY while Field Re-Sequencing is happening, up to, and including the time that the Initial Screen Display Phase is completed, the results will be disastrous.

4.9 DENTRY Program Phases

DENTRY was written in a structured manner. Each major function of program generation is coded in a separate 'included' module (PHASE). Great care was exercised in re-use of user data area, resulting in a compact DATABUS program.

Program DENTRY Phases:

DENTRY - Mainline
DENTRY1 - Input Screen
DENTRYC - Input User Comments
DENTRYE - Input Field Edit Definitions
DENTRYP - Print Screen Work Sheet
DENTRYR - Resequencing Fields
DENTRY2 - Generate Initial Screen Display
DENTRY3 - Generate Keyin Variables Sequence
DENTRYD - Generate Date-Check Sequence
DENTRYN - Numeric Field Classification
DENTRYT - Generate Table-Lookup Sequence
DENTRYR - Generate Range-Check Sequence
DENTRY4 - Generate Disk-Write Sequence
DENTRY5 - Generate Loop & End of Job Sequence
DENTRY6 - Final Merge

4.10 DENTRY Execution-Time Messages

Where the following messages are not self-explanatory within the context of the program phase that issues them, a brief description follows. If you must INTERRUPT DATASHARE while DENTRY is running, you will generally not lose anything done up to that point. Please read the Notes that follow if you are in doubt if an INTERRUPT is OK.

4.10.1 Phase DENTRY Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

Port Number must be in the first two data bytes! The users port number must be presented to DENTRY as a 'FORM 2' field.

Re-Start from a Prior Screen ? If you created a screen file from DENTRY before, you may use it again.

What is the Name of the Old Screen File ? Enter the name of the old screen-image file. If you used the DOS. NAME command to save your old screen file, be sure to specify the new name.

Source Program Name _____. Enter the name of the program that you want DSENTRY to generate.

Name Must be Different From Screen File! You cannot use 'SCREEN' as the first 6 letters of your program name. Please try to avoid using any of the DSENTRY Phase names also!

That File Exists Already ! Over-Write It ? If the source file name you specified already exists, this warning message is issued. If you respond 'Yes', the file you asked for will surely be destroyed.

(SOURCE) is being Prepared. This is just a comment.

(WORKNAME) is being Prepared. This is just a comment.

File Type: (S)equential or (I)sam ? The output of your program can either be Sequential, or ISAM. You could create a sample screen using Sequential output, then use the DOS. INDEX utility to index the file, then, re-start the screen and specify file type of ISAM.

4.10.2 Phase DSENTRY1 Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

SAMPLE ENTRY: NAME _____ LINE-LETTER (A-W) ? This message will be on the bottom line of the DATASHARE port while DSENTRY is ready to accept new screen lines. Respond with the Letter that corresponds to the vertical position on the screen that you want to enter a line.

Limit is 14 Variables on one Line. Self-Explanatory.

The SLASH character must be used as ___/___/__. Self-explanatory.

Are You Done (Y)es or (N)o ? If you tap the Enter key instead of

entering a valid Line Letter, DSENTRY will ask this question. Respond Yes or No.

4.10.3 Phase DSENTRYC Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

Enter Field Comments. Just a comment.

Comment for LINE (?) FIELD (?) (..... ...) Either the old comment (in case of re-start) will be displayed, or you will be expected to enter a comment. If you want the same comment as last time, just tap the Enter key.

4.10.4 Phase DSENTRYE Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

(R)equired Fields Field (T)ypes Field (P)rograms (N)o More Edits ?
This message shows the valid responses to the Level 1 Edit question. Not displayed is (C)LEAR - meaning Clear all Edits on a particular Line-Letter and Field number.

REQUIRED TYPE: (R)equired (F)ill-Controlled (B)oth -or- di(S)play-only ? Level 2 Edit message.

FIELD TYPE: (N)umeric (D)ate s(H)ift-key Inversion. Level 2 Edit message.

PROGRAM: (I)sam-Verify (T)able-Lookup r(A)nge-check. Level 2 Edit message.

(R)equired Field on Line-Letter ? Which Field _ ? Level 3 Edit message. Respond with the Line-Letter and Field number that correspond to the position on the screen that a particular edit is to be applied.

(F)ill-Controlled Field on Line-Letter ? Which Field _ ?

(B)oth Req & Fill Field on Line-Letter ? Which Field _ ?

DI(S)play-only Field on Line-Letter ? Which Field _ ?
(N)umeric Field on Line-Letter ? Which Field _ ?
(D)ate Field on Line-Letter ? Which Field _ ?
s(H)ift-key Inversion Field on Line-Letter ? Which Field _ ?
r(A)nge-Check Field on Line-Letter ? Which Field _ ?
(I)SAM-Lookup Field on Line-Letter ? Which Field _ ?
(T)able-Lookup Field on Line-Letter ? Which Field _ ?
(C)lear All Edit Field on Line-Letter ? Which Field _ ?

4.10.5 Phase DSETRYP Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

Do you want to print a Screen Work Sheet ? Respond Yes or No.
Make sure the printer is available.

I'M WAITING FOR THE PRINTER ! The printer is un-available.

Printing in Progress. Self-explanatory.

4.10.6 Phase DSETRYS Messages

NOTE: You may*NOT*INTERRUPT DATASHARE in this Phase.

Do you want the Same Sequence as Last Time ? If you performed a Re-Start, and if you did not change anything on the screen, you can respond 'Yes' if you want to maintain the same field sequencing as the last time you used the screen.

Do you want to Re-Sequence the Data Fields ? If new fields were entered on the screen, the sequencing information is lost. If you respond 'No' to this message, Fields will be sequenced Top-Down & Left to Right. If you

respond 'Yes', you will be asked the following question

Next Field is on Line-Letter ? Which Field ? Respond with the Line-Letter and Field number that corresponds to the field on the screen that you want the data-entry operator to enter next in sequence. Tap the Enter key when you have altered the sequence of all the fields necessary for your application. DSENTRY will then give the following message:

Are you Done ? If you have sequenced every field, respond 'Yes'. If you haven't sequenced every field, the remaining fields on the screen will automatically be sequenced to follow all the fields that you sequenced.

4.10.7 Phase DSENTRY2 Messages

NOTE: You may*NOT*INTERRUPT DATASHARE in this Phase.

Generating Initial Screen Display. Comment only.

Optimum Data Area is: _____ Bytes. DSENTRY accumulates the amount of User Data Area (UDA) that is necessary to execute your program. If the UDA requirements are less than 1000 bytes, certain techniques can be used in program generation that make data-entry easier. If the UDA is more than 1000 bytes, the following question will be asked:

IS THAT OK ? If you respond 'Yes', DSENTRY will use certain techniques in program generation. If you respond 'No', the following question will be asked:

Please...(Y/N) ? This is the last opportunity for you to let DSENTRY use the special techniques that make data-entry easier.

PROGRAM PHASE DSENTRY3 IS MISSING ***ABORT***. The program DSENTRY3/DBC is missing from your system.

4.10.8 Phase DENTRY3 Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

YOU ENTERED THIS PROGRAM INCORRECTLY ! Program DENTRY3/DBC cannot run by itself - it must be entered with certain variables from DENTRY/DBC.

Generating Keyin Statements.

ISAM Lookup on Line (?) Var.# (?) Is a HIT (G)ood or (B)ad? If the result of an auxillary ISAM lookup is required before the data-entry operator may proceed, respond 'Good', else, respond 'Bad'.

4.10.9 Phase DENTRYD Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

(No messages).

4.10.10 Phase DENTRYN Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

Enter Decimal Format for Line (?) Var.# (?) _____. Enter the correct DATABUS numeric format for a field. For example, if a numeric field is 8 bytes long, a valid format is 5.2, or 4.3, or 6.1, etc. The decimal point actually occupies one byte of the field.

4.10.11 Phase DENTRYT Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

How Many Values for Table Lookup on Line (?) Var.# (?) _____. Enter the number of values that are to be in the table.

Is a Match Considered (G)ood or (B)ad ? If you respond 'Good', the data-entry operator must enter a value that is IN the table. If you respond 'Bad', the data-entry operator must enter a value that is NOT IN the table.

NEXT TABLE VALUE: Enter the next value that is to be in the table.

That Value has Been Rejected! You entered an incorrect (or null) value.

4.10.12 Phase DSENTRYR Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

Enter the Lowest Value for Range-Check on Line (?) Var (?). If the lowest possible value for a field is '150', enter exactly that.

Enter the Highest Value for Range-Check on Line (?) Var (?). If the highest possible value for a field is 'XXXX', enter exactly that.

4.10.13 Phase DSENTRY4 Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.

Generating Disk-Write Sequence. Comment only.

Prime ISAM-Key is on Line-Letter (?) Which Field (?). The field on the Line-Letter and at the Field number you define here will be the key by which records will be added to your ISAM database.

4.10.14 Phase DENTRY5 Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.
Generating End of Job Sequence. Comment only.

4.10.15 Phase DENTRY6 Messages

NOTE: You may INTERRUPT DATASHARE any time in this Phase.
FINAL MERGE. Comment only.

If you want to Save your SCREEN FILE for Another Pass: IT'S NAME
IS: _____. Comment only.

CHAPTER 5. PRINT PROGRAM GENERATOR - DSPRINT

DSPRINT is a DATABUS program which provides for efficient report generation and data printing capability for a DATASHARE 3 or DATABUS 11 system. Using this program, a data-entry screen designed by the DATABUS program DSENTRY is analyzed, and a DATABUS program is generated to print the output file of the program generated by DSENTRY. The output of DSPRINT is a DATABUS source file that can be directly compiled with the DOS. DATABUS (DBCMP) compiler, or, using the DOS. EDIT command, additional complex functions can be inserted by the programmer.

Many data-reporting programs are simply 'iterative' loops, i.e.,

1. Reading information (input);
2. Performing calculations (processing);
3. Recording Results (output).

The programmer's task is similar, no matter which data reporting task is assigned to him. DSPRINT will allow the programmer to easily create a functional DATABUS program, since all that is necessary to accomplish program generation with DSPRINT is defining report headings and sub-total (Level) breaks.

Simple data-reporting programs can be created in a matter of minutes; complex programs take somewhat longer. Since the output of DSPRINT is a standard DATABUS source file, any changes that the programmer wishes to make to the program generated by DSPRINT can be performed using the DOS. EDIT command.

5.1 Input to DSPRINT

The input to DSPRINT is the "SCREEN" file created by DSENTRY, and a programmer, sitting at a DATASHARE Terminal.

5.2 Output of DSPRINT

The output of DSPRINT is a program source file. DSPRINT will automatically create source code for:

1. ISAM file opening sequence;
2. Sequential file opening sequence;
3. Read database sequence;
4. Check for "LEVEL BREAKS" sequence;
5. Accumulate Sub-Totals;
6. Print Report Headings;
7. Print Detail Lines;
8. Print Sub-Totals;
9. Print Grand-Total;
10. File-closing sequence;
11. End of job sequence.

5.3 Creating a Program Source File

The program that will be created by DSPRINT is a standard 'text' file. You may edit the file, compile it, list it, etc.

The program generated by DSPRINT needs a name. Program names must conform to the DOS standard naming conventions, i.e., up to 8 letters and numbers with no special characters or spaces embedded.

5.3.1 ISAM or Sequential Input Files

The input disk file that your DSPRINT generated program will create can either be an ISAM file, or a sequential file. ISAM files are created by executing the INDEX utility program. Direct files (including sequential access) are standard Datapoint space-compressed files. All of the variables included in a screen layout (with the exception of "Display-Only" variables) will be read from a single record in the order in which they appear on the screen.

DSPRINT will now ask:

WHAT IS THE NAME OF THE SCREEN-IMAGE FILE CREATED BY PROGRAM
DSENTRY ?

Respond with the name of the file name created by DSENTRY.

DSPRINT will now ask:

SOURCE PROGRAM NAME _____.

Respond with the name of the program that you want DSPRINT to create.

DSPRINT will now ask:

INPUT FILE TYPE:(S)SEQUENTIAL OR (I)SAM ?

Respond with the letter 'S' or 'I' as necessary. The input file type should be the same type as the output file type created by DSENTRY. If you aren't sure which file type, consult the DOS DATABUS (DBCMP) User's Guide for a discussion of the different file structures.

5.4 Creating the Print Program

At this point, the screen-image file created by DSENTRY will be displayed. The letters A thru W will be displayed down the left side of the screen. Each letter represents one line. For instance, the letter 'C' represents line three (3). The bottom line of the screen will be used for further machine-programmer communications. Refer to Appendix A for a sample screen.

5.4.1 Field Elimination

DSPRINT will ask you if you want to eliminate any of the fields that are on the screen from the printed report.

DO YOU WANT TO ELIMINATE ANY FIELDS FROM THE PRINTOUT ?

If all the data on the screen is to be printed, respond 'No'. Otherwise, you will be asked the following question:

ELIMINATE FIELD ON LINE-LETTER ? _

To eliminate a field on the screen, enter a valid letter (A thru W). DSPRINT will now ask:

WHICH FIELD (1-??) ?

Respond with the field number on the selected line that is to be suppressed in the final printout.

5.4.2 Re-Instating an Eliminated Field

If you change your mind about eliminating a field from the printout, just ask for the same Line-Letter and Field Number. DSPRINT will now ask:

RE-INSTATE FIELD (??) on line (?) ?

Respond Yes or No as appropriate.

Refer to the sample Screen-Image in the Appendix A.

Line-F contains operator prompts and input variables for:

C/O _____

In some reports, this information may be meaningless. To eliminate it from the printout, tell DSPRINT to eliminate Line-Letter (F), Field (1) and Line-Letter (F), Field (2).

When you have eliminated all un-necessary fields from the printout, tap the Enter key. DSPRINT will now ask:

ARE YOU DONE (Y)ES OR (N)O ?

A response of 'Yes' will tell DSPRINT that no more fields are to be eliminated from the printout.

If the length of all the variables left after field elimination is greater than 254 bytes, DSPRINT will display:

YOUR RECORD IS nnn BYTES TOO BIG.

If this occurs, your record is simply too large for this

program to accomodate. Generally, 249 bytes is a good design maximum for data files.

5.4.3 User Comments

A single screen line can have as many as 14 fields on it, and there can be up to 23 lines of screen data. It would be undesirable to generate a program of such size without having some kind of comments associated with each of the fields. Therefore, DSPRINT will comment the generated program with the user comments entered by the programmer in program DSENTRY.

5.4.4 Control Groups (Level Breaks)

Total calculation and output are normally performed on data accumulated for a group of related records which form a control group. When the fields of a record which determine the control group change, a control break occurs indicating a new control group is starting. When a break occurs (shown by control level indicators being turned on), calculation and output operations are performed using information accumulated from all records in the previous control group. The control group that changes most frequently is referred to as the lowest level (L1), the next control group change is (L2), etc. DSPRINT allows for three Level Breaks (Minor, Intermediate, and Major).

5.4.4.1 Defining Level Breaks

If sub-totals are to be accumulated and printed, respond to the following DSPRINT question:

LEVEL-1 BREAK IS ON LINE-LETTER ? WHICH FIELD ?

If no sub-totals are to be accumulated, simply tap the Enter key when the preceeding question is asked. The generated program will print a simple listing of the data.

DSPRINT will ask for the LEVEL-2 Break and the LEVEL-3 break in the same manner. You don't need either LEVEL-2 or LEVEL-3 breaks if you don't want them. If you do specify level breaks, the highest numbered level is the major total, the next lower number is intermediate, and the lowest number is the minor total.

5.4.4.2 Accumulating Sub-Totals

When Level Breaks are specified, counts, or arithmetic sums can be performed on a data field. This field is not always the same field as the Level-break field. For instance, if Invoice Date changes, you may want to accumulate the Invoice Totals (certainly not the dates!).

5.4.5 Sequence of Fields in Printout

Since the purpose of DSPRINT is to automate program generation as much as possible, certain assumptions are made about the final report. One of the assumptions is that the data is to be printed in the same sequence as it was entered (as defined by DSENTRY). If you decide that the printed output should be in a different sequence from the defined sequence, perform the following steps:

1. Save your Screen File (use the DOS. COPY command);
2. Execute DSENTRY;
3. Perform a "Re-Start" on the Screen file;
4. Re-Sequence the fields into the new order;
5. After DSPRINT has been run, restore the Screen file.

5.4.6 Report Headings

In order for the printed output of the DSPRINT generated program to be of any value to the user of the report, descriptions of the data included on the report are necessary.

5.4.6.1 Page Heading

The DSPRINT generated program will allow up to 78 bytes of data to be printed every time that a page "overflow" occurs. This heading should generally describe the contents of the report. DSPRINT will automatically include the DATABUS code to print the date and page number on the same line as the page heading.

5.4.6.2 Field Heading

Above each field that is printed, a brief description of the contents of that field will be printed. DSPRINT will display a "form" that you can enter a field heading into.

CHAPTER 6. UPDATE PROGRAM GENERATOR - DSUPDATE

DSUPDATE is a DATABUS program which provides for efficient data update program generation capability for a DATASHARE 3 or DATABUS 11 system. Using this program, a data-entry screen designed by the DATABUS program DSENTRY is analyzed, and a DATABUS program is generated to update the data file of the program generated by DSENTRY. The output of DSUPDATE is a DATABUS source file that can be directly compiled with the DOS. DATABUS compiler, or, using the DOS. EDIT command, additional complex update functions can be inserted by the programmer.

Many data-updating programs are simply 'iterative' loops, i.e.,

1. Retrieve Variables from Data Base;
2. Keyin Changed Variables;
3. Edit Variables;
4. Update Variables on Data Base;

The programmer's task is similar, no matter which data update task is assigned to him. DSUPDATE will allow the programmer to easily create a functional DATABUS program, since all that is necessary to accomplish program generation with DSUPDATE is defining any necessary range-check and table-lookup values.

Simple data-entry programs can be created in a matter of minutes; complex programs take somewhat longer. Since the output of DSUPDATE is a standard DATABUS source file, any changes that the programmer wishes to make to the program generated by DSUPDATE can be done using the DOS. EDIT command.

6.1 Input to DSUPDATE

The input to DSUPDATE is the "SCREEN" file created by program DSENTRY, and a programmer sitting at a DATASHARE Terminal.

6.2 Output of DSUPDATE

DSUPDATE will automatically create source code for:

1. ISAM file opening sequence;
2. Sequential file opening sequence;
3. Open auxillary ISAM "lookup" files;
4. In-line date verification;
5. In-line range-checking of variables;
6. In-line table-lookup validation;
7. Reserved (display-only) variables;
8. File-closing sequence;
9. End of job sequence.

6.3 Creating a Program Source File

The program that will be created by DSUPDATE is a standard 'text' file. You may edit the file, compile it, list it, etc.

DSUPDATE will ask for the name of the SCREEN IMAGE FILE created by DATABUS program DENTRY.

The program generated by DSUPDATE needs a name. Program names must conform to the DOS standard naming conventions, i.e., up to 8 letters and numbers with no special characters or spaces embedded.

6.4 ISAM or Sequential Input Files

The input disk file that your DENTRY generated program created can either be an ISAM file, or a sequential file. ISAM files are created by executing the INDEX utility program. Direct files (including sequential access) are standard Datapoint space-compressed files. All of the variables included in a screen layout (with the exception of "Display-Only" variables) will be read from a single record in the order in which they appear on the screen.

DSUPDATE will now ask:

INPUT FILE TYPE:(S)EQUENTIAL OR (I)SAM ?

Respond with the letter 'S' or 'I' as necessary. If you

aren't sure which file type, consult the DOS DATABUS (DBCMP) User's Guide for a discussion of the different file structures. In any case, this your response to this question should match your response to a similar question in DSENTRY.

If you respond 'Isam', you will later be asked the following question about the Prime ISAM key:

PRIME ISAM-KEY IS ON LINE-LETTER (?) WHICH FIELD (?)

The variable at that location will be used to retrieve and update the records of your data base.

6.5 User Comments

A single screen line can have as many as 14 fields on it, and there can be up to 23 lines of screen data. It would be undesirable to generate a program of such size without having some kind of comments associated with each of the fields. DSUPDATE will comment the generated program source file with the comments entered by the programmer while running DSENTRY.

6.6 Field Edits

Most data-update programs require some type of field validation. For example, an employee number that has to be verified before payroll data can be entered might require an access to the master employee database. In the case of a date, correct month, day, and year must be checked.

DSUPDATE will use the Field Edits as defined by the programmer in DSENTRY.

6.6.1 Auxillary ISAM Verify

The ISAM verify code generated by DSENTRY will just be a simple ISAM read into a 1 byte work area. If a real database is to be used for field validation, it is the programmer's responsibility to include the correct read statement in the generated source program.

6.6.2 Table-Lookup

If the programmer requested that a table-lookup be performed on a field, DSUPDATE will ask for the number of entries in the table. For instance, a part number prefix might be 'AA' or 'BB' or 'CC' - if it isn't, the operator must re-key the field. On the other hand, entering a code of 'Y7' or 'W3' might be invalid, in which case, the operator must re-key the field. DSUPDATE will need to know if the resulting lookup is good (meaning the data must be present in the table) or bad (meaning the data must not be present in the table). Now, DSUPDATE will accept the actual table values from the programmer.

6.6.3 Range Check

If the programmer requested that a Range-Check be performed on a field, DSUPDATE will ask for the lowest and highest values permitted. No attempt is made to see that the two values are arithmetically correct. It is entirely up to the programmer to assign good upper and lower limits. Range checks can be performed on alphameric or numeric fields.

6.6.4 Data Entry Sequence

DSUPDATE will use the sequence information from the last time the screen was used to create the current program.

6.7 DSUPDATE Program Phases

DSUPDATE was written in a structured manner. Each major function of program generation is coded in a separate included module (Phase). Great care was exercised in re-use of user data area, resulting in a compact (less than 700 bytes) DATASHARE program.

Program DSUPDATE Phases:

Phase DSUPDATE - Mainline
Phase DSUPDD - Generate Date-Check Sequence
Phase DSUPDT - Generate Table-Lookup Sequence
Phase DSUPDR - Generate Range-Check Sequence

Program DSUPDAT1 Phases:

Phase DSUPDAT1 - Misc Routines & Final Merge

APPENDIX A. SAMPLE DSGEN SCREEN LAYOUT

The letters on the left margin represent the screen line.

A ACME ARMADILLO MEAT SPECIALTIES CORPORATION
 B INVOICE # _____ INVOICE DATE ___/___/___
 C -SOLD TO- -SHIP TO-
 D Name _____
 E Address _____
 F c/o _____
 G City _____
 H
 I Item Qty Description
 J _____
 K _____
 L _____
 M _____
 N
 O Invoice Total _____
 P
 Q
 R
 S
 T
 U
 V
 W

APPENDIX B. SAMPLE DSPRINT REPORT (DETAIL LIST)

ACME CORP. DAILY SALES REPORT

9/02/76 PAGE 1

Inv.#	Customer Name	Date	Amount
122231	AAA Specialties	90276	100.00
122232	AAA Specialties	90276	50.00
122233	ABC Industries	90276	125.35
122234	ABC Industries	90276	4.00
			279.35 ****

APPENDIX C. SAMPLE DSPRINT REPORT (ONE LEVEL BREAK)

ACME CORP. SALES RECAP BY CUSTOMER

9/04/76 PAGE 5

CUSTOMER NAME	DATE	ITEM	QTY
GRANNY'S MEAT CO.	72676	104	100
			100*
GREX ARMADILLO CO.	83176	102	3
GREX ARMADILLO CO.	90176	102	5
			8*
HAPPY MEAT PROD'S.	81976	103	1
HAPPY MEAT PROD'S.	81976	104	21
			22*
			130****

APPENDIX D. DSPRINT GENERATED PROGRAM LOGIC FLOW

The program generated by DSPRINT follows a strict format. The format is intended to follow DATAPOINT RPG II program logic as closely as possible.

1. File Opening Sequence
2. Read a Record from the file just processed
3. Test for End of File
4. Turn On 'Level Indicators'
5. See if Matching Fields Specified
6. Turn on Record Ident Indicators
7. Test for Control Break
8. Bypass Total Calcs if First Cycle
9. Perform Total Time Calculations
10. Perform Total Time Output
11. Test Last Record Indicator
12. Test Page Overflow Indicator
13. Move Data from Input Area to Fields
14. See if Matching Record
15. Perform Detail Time Calculations
16. Perform Detail Time Output
17. Turn off all Indicators
18. Go to Read another Record
19. Page Heading Routine
20. End of Job Routine

APPENDIX E. SCREEN-IMAGE WORK FILE DEFINITION

- . THE SCREEN-IMAGE WORK FILE CONTAINS FIVE LOGICAL FILES:
- . 1. SCREEN-IMAGE LINE RECORDS 1-24 BYTES 1-81
- . 2. FIELD-EDIT DEFINITIONS RECORDS 1-24 BYTES 82-249
- . 3. USER COMMENTS RECORDS 25-72 BYTES 1-248
- . 4. FIELD SEQUENCE INFORMATION RECORDS 73-74 BYTES 1-249
- . 5. FIELD HEADINGS (DSPRINT ONLY) RECORDS 75-189 BYTES 1-249

SCREEN FILE SCREEN-IMAGE WORK FILE

*SCREEN-IMAGE RECORDS.....

. ALGORITHM FOR SCREEN-IMAGE RECORD NUMBER:
 . RRN = LETTER (A THRU W) TRANSLATED TO (1 THRU 23)

.FIELD	TYPE	LENGTH	FROM-TO	DESCRIPTION
-----	----	-----	-----	-----
CRTLET	DIM	1	1-1	LINE-LETTER
AWK78	DIM	78	2-79	SCREEN-IMAGE LINE
TBNVAR	EQU		80	TAB TO # OF VARIABLES
NVAR	FORM	2	80-81	# OF VARIABLES THIS LINE

*FIELD-EDIT DEFINITIONS.....

. EACH OF THE 14 POSSIBLE VARIABLES ON A LINE HAS ALL THE
 . FOLLOWING VARIABLES ASSOCIATED WITH IT.
 . VALUES START AT POSITION 82, AND CONTINUE THRU POSITION 249
 . INCLUSIVE.

. ALGORITHM FOR CALCULATING READ/WRITE TAB VALUE FOR FIELD EDITS
 . TABVAL=(CURRENT VARIABLE-1)*(EDIT LENGTH)+(STARTING POSITION)

HORIZ	FORM	" 0"	HORIZONTAL POSITION OF VARIABLE
LENGTH	FORM	" 0"	LENGTH OF VARIABLE
VARNUM	FORM	" 0"	VARIABLE NUMBER
EDIT1	INIT	" "	REQUIRED
EDIT2	INIT	" "	TYPE
EDIT3	INIT	" "	PROGRAM
EDIT4	INIT	" "	WORK AREA
EDIT5	INIT	" "	WORK AREA

. -----
 . 12 = EDIT LENGTH


```

. NOTE: EDIT4 & EDIT5 ARE USED AS WORK AREAS AS FOLLOWS:
.     EDIT4      "2"==>FIELD IS TO BE PRINTED ON LINE 2
.     EDIT5      "*"==>FIELD IS NOT TO BE PRINTED
*.....FIELD HEADING INFORMATION.....
. IN ORDER FOR THE GENERATED PRINT PROGRAM TO CREATE A
. MEANINGFUL REPORT, FIELD HEADINGS ARE NECESSARY.
. THE SAME ALGORITHM USED FOR USER COMMENTS APPLIES TO
. FIELD HEADINGS.
HEDRRN  FORM      " 75"      RECORD NUMBER
HEDTAB  FORM      3          COMPUTED READ/WRITE TAB
.....

```

INDEX

CLEAR 4-9, 4-16
COMMENT 4-5, 4-6, 4-7, 4-15, 4-16, 4-18, 4-20, 4-21, 5-5, 6-3
COMMON 2-2, 2-3
DATE 2-1, 2-2, 4-1, 4-4, 4-5, 4-6, 4-7, 4-9, 4-13, 5-6, 6-2, 6-3
DSGEN 1-1, 2-1, 2-2, 2-3, 3-1
DSENTRY 1-1, 2-2, 4-1, 4-2, 4-3, 4-4, 4-5, 4-6, 4-7, 4-8, 4-9,
4-10, 4-11, 4-12, 4-13, 4-14, 4-15, 4-16, 4-18, 5-1, 5-2, 5-3,
5-5, 5-6, 6-1, 6-2, 6-3
DSPRINT 1-1, 2-2, 5-1, 5-2, 5-3, 5-4, 5-5, 5-6, 5-7
DSUPDATE 1-1, 6-1, 6-2, 6-3, 6-4
EDIT 1-1, 4-1, 4-2, 4-4, 4-5, 4-6, 4-7, 4-8, 4-11, 4-14, 4-16,
4-17, 5-1, 5-2, 6-1, 6-2
FILL 4-8, 4-16
FORM 1-1, 4-10, 4-14, 5-5, 5-7
INTERRUPT 4-13, 4-14, 4-15, 4-16, 4-17, 4-19, 4-20, 4-21
ISAM 4-1, 4-2, 4-5, 4-10, 4-11, 4-15, 4-19, 4-20, 5-2, 6-2, 6-3
LEVEL 4-8, 4-9, 4-16, 5-1, 5-2, 5-5, 5-6
MASTER 1-1, 2-3, 4-7, 6-3
PORT 1-1, 2-2, 4-3, 4-14, 4-15
PRINT 1-1, 4-14, 4-17, 5-1, 5-2, 5-5, 5-6
PROMPT 3-1, 4-4
RE-INSTATE 5-4
REQUIRED 2-2, 4-7, 4-9, 4-16, 4-19
SCREEN 1-1, 4-1, 4-2, 4-3, 4-4, 4-5, 4-6, 4-7, 4-9, 4-10, 4-11,
4-12, 4-13, 4-14, 4-15, 4-16, 4-17, 4-18, 4-21, 5-1, 5-2, 5-3,
5-5, 5-6, 6-1, 6-2, 6-3, 6-4
SEQUENCE 1-1, 4-1, 4-2, 4-12, 4-13, 4-14, 4-17, 4-18, 4-20,
4-21, 5-2, 5-6, 6-2, 6-4
SEQUENTIAL 4-1, 4-2, 4-15, 5-2, 6-2
SLASH 4-4, 4-5, 4-15
TYPEWRITER 4-10
TYPE 3-1, 4-2, 4-5, 4-7, 4-8, 4-15, 4-16, 5-3, 6-3

Manual Name _____

Manual Number _____

READER'S COMMENTS

Did you find errors in this manual? If so, specify by page.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____

All comments and suggestions become the property of Datapoint.

Fold Here

Fold Here and Staple



FIRST CLASS
Permit
5774
San Antonio
Texas

BUSINESS REPLY MAIL
No Postage Necessary if mailed in the United States

Postage will be paid by:

DATAPPOINT CORPORATION
DIRECTOR OF SOFTWARE SUPPORT
8550 Datapoint Drive, Mail Station# N60
San Antonio, Texas 78284

