Burroughs

B 20 Data Manager

PROGRAMMER'S MANUAL

(Relative to Release Level 2.0)

PRICED ITEM

Printed in U.S.A. May 1983 1154333

Burroughs

B 20 Data Manager

PROGRAMMER'S MANUAL

(Relative to Release Level 2.0)

PRICED ITEM

Printed in U.S.A. May 1983 1154333

The names, places, and/or events depicted herein are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association is purely coincidental and unintentional.

The within information is not intended to be nor should such be construed as an affirmation of fact, representation or warranty by Burroughs Corporation of any type, kind, or character. Any product and related materials disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed license agreement. The only warranties made by Burroughs with respect to the products described in this material are set forth in the above mentioned agreement.

The customer should exercise care to assure that use of the application package will be in full compliance with laws, rules and regulations of the jurisdiction with respect to which it is used.

The information contained herein is subject to change. Revisions may be issued from time to time to advise of changes and/or additions.

Any comments or suggestions regarding this document should be forwarded to Documentation, Systems Development Group, Burroughs Corporation, 3519 West Warner Avenue, Santa Ana, CA, 92704.

TABLE OF CONTENTS

INTRODUCTION

This manual is directed to programmers who will create application programs with the 2.0 release of B20 Data Manager. B20 Data Manager does not require knowledge of programming languages but it does assume you have a knowledge of data processing concepts and terminology. You can create programs quickly and easily by using Data Manager (DM) programs and forms.

The manual contains an overview of the system as well as an outline of its structure. It also discusses the limitations of the system. In addition, you are shown how to create your own applications. There are operating instructions for the two major programs, DMCREATE and DMRUN.

Explanations of the DMCREATE forms and the prompts involved with each form are found in Appendix A - How to Understand Forms. Error messages are listed and explained in Appendix B - DMCREATE Error Messages, in Appendix C - DMRUN Error Messages, and Appendix D - DMPLIST Error Messages.

Easy-to-follow installation instructions are found in Section 3. With these instructions, you should have no trouble installing B20 Data Manager.

Appendix E describes how to convert programs from release level 1.0 or 1.1 to release level 2.0.

Other manuals which may be helpful to you are:

- B20 Data Manager Application User's Manual, form 1154366.
- B20 Data Manager Training Guide, form 1154325.
- B20 Operations Training Course (See Part II, Learning to Use the System Software), form EL 6300.

- B20 System Software Operations Guide, form 1154242.
- B20 Systems Forms Reference Manual, form 1148715.

The style identification for B20 Data Manager is B20 DMR.

SECTION 1

OVERVIEW AND STRUCTURE OF B20 DATA MANAGER

This section presents an overview of B20 Data Manager and a description of the B20 Data Manager system structure.

SYSTEMS OVERVIEW

Burroughs B20 Data Manager (DM) is a practical aid for developing application programs for Burroughs B20 computers. As an applications programmer using B20 Data Manager, you will find this system increases your productivity because you spend less time on the routine tasks of software development.

After creating a Data Manager application system, you can modify and extend it to meet the needs of a particular application user. You can do this quickly and easily, without the problems presented by traditional programming. B20 Data Manager helps you produce marketable, professional-appearing software.

While B20 Data Manager is primarily designed to be used by experienced programmers, any person who understands elementary data processing concepts can learn to create programs using B20 DM. Using B20 Data Manager is easier than learning a programming language because it uses a series of menus and simple commands. You can concentrate on defining a particular business application rather than technical programming concepts.

You can use Data Manager's Standard Forms quickly and easily. These forms are fully automatic and do not require any format specification. If you need especially complex forms or forms compatible with existing forms design standards, you can create them with another Burroughs product, the Forms Editor. The forms which you create with this product (Forms Editor forms) can interface to the B20 DM programs.

Up to five indices can be maintained automatically for each application file. Inquiries and reports can then access files in various sequences without sorting. DM programs can reference a maximum of nine application files. This number may be lower at some installations due to hardware configuration and other factors determined by your particular applications.

B20 Data Manager allows you to use 28 standard field descriptions. They describe alphanumeric and numeric fields of various types and lengths. You can specify additional field descriptions for your particular needs.

DM programs can be linked dynamically to other programs. You can define the sequence in which programs will be executed to establish an automatic job flow. An operator can vary that sequence at run time. B20 DM has 45 registers which can be used to pass information between linked DM programs.

Data Manager supplies 12 simple BASIC-like constructs for function statements. The function statements give a concise means of defining computations, verifying data, and defining record selection criteria.

PROGRAM TYPES

B20 Data Manager offers many advantages and a wide range of features in one package. The design of B20 DM can help you implement complete application systems in a short time. You can create the following program types using B20 DM:

1. Menu Program.
2. File Maintenance Program.
3. Batch Data Entry Program.
4. Batch Update Program.
5. Real-time Update Program.
6. Report Program.
7. Inquiry Program.
8. Run-time Data Program.

Menu Program

You can use Data Manager Menu programs to initiate other DM programs and programs written in any of the languages available on the B20. The DM Menu Program allows up to 36 menu selections. One option of this program type allows additional data entries other than the menu selection at run time. Another option is to create a file which can contain a variety of user-specified information. For example, the file could store a history of programs run on the system.

In addition, you can specify up to five indices for the file created by the menu.

File Maintenance Program

The File Maintenance program type has several important features. It allows the creation and maintenance of a single disk file. Records can be added, changed, or deleted from this file.

A File Maintenance program can reference up to eight additional files plus the file being maintained. The program can reference these files for simple existence checks, or as sources of information. The system can then display the information for visual verification, or the information can appear as data in an entry field.

Maintenance programs provide two opportunities to use function statements. Initial Functions are performed only once, at the beginning of the program before any forms are displayed. These statements can be used to reference information passed from the previous program. Repeating Functions can be used to specify computations and additional data validation tests. Repeating Functions are performed every time the form is processed. This will happen after all the fields on the form have been entered but before the record is written.

There are also many important operational features in Maintenance programs. The Skip-to-Field entry specifies that several fields can be skipped when a given field is left blank. The Field Duplication entry indicates that information in one field can appear as a default for the same field when the form is repeated. The Automatic Increment specifies that a constant amount be added to a field each time a form is repeated.

In addition, the File Maintenance program has several data validation features. The program can reference a file to perform an existence check. You can specify Mandatory Entry fields (fields in which the application user must enter data). You can apply up to five range checks on one entry. There is a maximum of 20 numeric and 20 alphabetic ranges that can be specified in a single program. Each range can be applied to any number of fields.

The Allow Entry attribute marks those fields in which the application user can enter or change data. If an entry is not allowed, the field is used only to display data for visual validation.

Batch Data Entry Program

The data validation features of a Maintenance program allow the B20 programmer to create Batch Data Entry programs as well. The Maintenance program can create a batch transaction file.

Batch Update Program

In Batch Update, transactions are read from a batch transaction file and used to update, add records to, or delete records from as many as eight secondary files. The transaction file itself can also be updated. Computed items as well as transaction data can be used in the update. Function statements are used to specify update computations.

Real-time Update Program

Real-time Update programs include all of the data validation and operational features of File Maintenance programs. You can concurrently update or add records to as many as eight files. These files can be used for additional input and also for existence checking. Function statements are used to specify update computations.

In addition, Real-time Update can create a log file of the transactions entered. This log file could be used as input to a Report program or a Batch Update program if the transactions required further processing.

Report Program

Various kinds of reports can be created with B20 Data Manager. You can specify a simple format, such as an ordinary list of records. You can create more complex documents, too, because Data Manager can provide three levels of grouping of output items and can produce up to 16 subtotals and totals.

You can specify any of several optional report elements in addition to the report body. These optional report elements include a report header or trailer, page and column headings, page trailers, and special continuation page information for multiple-page, pre-printed forms.

You can select records by specifying a simple range of records or by complex combinations of conditions. Function statements can be used to include computed items in the report, as well as to calculate simple statistics. Also, you can control the report sequence by sorting the control file.

You have complete control over formatting and output of reports. You can easily merge information into many existing pre-printed forms. Report output can be written to the spooler for cluster operations, directly to a local printer for pre-printed forms and stand-alone operations, to the screen for immediate viewing, or to a disk file.

Inquiry Program

Inquiry provides a less complex method of reporting data. When the Inquiry program is run, specified data is displayed on the screen. You can include computed items in the inquiry, as well as information from up to nine files.

Run-time Data Program

The Run-time Data program gives an easy, flexible method to enter run-time data (alpha and/or numeric constants) for a succeeding program. You also can enter calculation constants for batch and real-time update in a Run-time Data program. Run-time data is frequently used to specify a range of records when printing a report.

SYSTEM STRUCTURE

You will be working primarily with four components of the B20 Data Manager system:

1. DMCREATE.
2. DMRUN.
3. Application Program Libraries.
4. Forms Libraries.

You can create and maintain B20 Data Manager programs with DMCREATE. Programs created with DMCREATE are then run using DMRUN.

Application programs created with DMCREATE are stored in one of several Program Libraries. Each Program Library can contain an initial menu and one or more application programs. DMRUN allows ten different Program Libraries. You supply the Program Libraries to the application user.

If your programs use Forms Editor forms, you will need to consolidate these forms into Forms Libraries by using the Librarian utility. All the forms used by programs in a single Program Library must have their forms in a single Forms Library.

SECTION 2

INTRODUCTORY DATA MANAGER PROGRAMMING CONCEPTS

To learn how to create and run Data Manager programs as easily as possible, you should read the next few pages on Data Manager concepts. This section of the manual will provide a good introduction to the terminology used on DMCREATE's programming forms and in other sections of this manual. The concepts are presented in the following order:

1. Entry forms.
2. Program help, field help, and error help forms.
3. Forms Libraries.
4. Program Libraries - initial menu.
5. Application data bases.
6. Record structures, Standard Descriptions, Descriptions.
7. File Management.
8. Maintenance Program.
9. Variable names, Sources.
10. Program linkage.

ENTRY FORMS

B20 Data Manager programs are either interactive or batch. Interactive programs require data entry forms, i.e., screen images on which data is entered. Batch programs do not require entry forms.

Interactive programs which include:

a. Maintenance programs.
b. Real-time Update programs.
c. Inquiry programs.

d. Run-time Data programs.
e. Menu programs.

Batch programs which include:

a. Batch Update programs.
b. Report programs.

Interactive programs require the application user to make entries on a form and press function keys, and therefore require a data entry form. The batch programs, however, process without operator interaction and do not require an entry form.

When you create an interactive program, you have a choice of two types of entry forms:

1. A Forms Editor form.
2. A Standard form.

FORMS EDITOR FORM

A Forms Editor form is created by the Forms Editor utility and allows you almost complete freedom in form layout and highlighting.

STANDARD FORM

A Standard form has the advantage that you do not have to create a separate form at all. DMRUN instantly generates an entry form image from the specifications in your program each time it is run. For complete information on forms, refer to the Forms Interface section of this manual.

HELP FORMS

When you begin to create Data Manager programs, you will notice many opportunities to specify help form names. All program types offer the option of specifying at least one of the following:

1. Program help forms.
2. Entry help forms.
3. Error help forms.

These forms are an important part of a finished application system. They allow you to expand Data Manager's error messages with useful solutions for operational problems.

Default help forms are provided with Data Manager, but your own help forms can be created with the Forms Editor utility. You can create DM programs without specifying any help form names and easily add them later.

Complete information on errors and help forms can be found in the Error Handling and Forms Interface sections of this manual.

FORMS LIBRARIES

The entry forms and help forms for your programs are stored in a Forms Library. There can be a separate Forms Library for each Program Library. The Forms Library file ID is stored in the Program Library control record. When you create a Program Library with DMCREATE, you have an opportunity to enter the Forms Library file ID.

If your entry forms are all Standard forms and you do not use any help forms, you do not need to specify a Forms Library file ID. If you do specify a Forms Library file ID, the Forms Library is required in order for DMRUN to initiate any program in the Program Library.

More information on Forms Libraries is in the Forms Interface section of this manual.

PROGRAM LIBRARIES

Data Manager programs are created and stored in Program Libraries. A Program Library is an Indexed Sequential Access Method (ISAM) file that contains all the DM programs for a particular application. All inventory control programs, for example, should be in the same Program Library.

After you have created a new program in a Program Library, the easiest way to test it is to use the Direct Program Selection feature of DMRUN. This allows you to run any program in any Program Library by entering the library file ID and the program name.

BUILDING AND INSTALLING A PROGRAM LIBRARY

Use the following process to save time in selecting programs when you already have several programs in a Program Library. If you use DMCREATE to create a new Program Library, enter an initial DM program name. This is usually the name of a Menu program which provides access to all other programs in the library.

You can use a utility feature of DMRUN to install Program Libraries in the DMRUN Library Selection menu. After a Program Library is installed, a description of the application appears on the DMRUN Library Selection menu each time DMRUN is executed.

When you have specified an initial DM program for the library and installed the library in the DMRUN Library Selection menu, programs can be selected entirely by menus. The application user does not need to remember any program or library names. He or she simply selects an application description from the Library Selection menu. DMRUN automatically loads and runs the initial program in the corresponding library. This program, as mentioned above, should be a menu which provides access to all the other programs in the application system.

APPLICATION DATA BASES

The B20 System software provides a very efficient and powerful ISAM file capability. All application data files accessed by Data Manager are ISAM files. To permit the use of multiple-file data structures in your application data bases, Data Manager offers a primary-secondary file feature.

PRIMARY AND SECONDARY FILES

A Data Manager Primary file can be characterized as a "file" which is processed "first" and which can contain key information for accessing other secondary files. The words "file" and "first" are in quotation marks because their meanings vary a little depending on the type of Data Manager program being considered. In a Real-time Update Program, for example, the fields on the form are considered to be the primary file.

Suppose your application data base contains a customer file and a salesman file. The customer file records contain, for example, a salesman number which allows Data Manager to access a specific salesman record. You could create a Maintenance program that would maintain the customer file as a primary file and either verify data with the salesman file or extract information from it as a secondary file.

To incorporate greater flexibility in the primary-secondary file relationship, secondary files can be given a critical attribute. If a secondary file is critical and an attempt to access that file results in a record-not-found condition, DMRUN will display an error message for the application user. If the file is not critical, the transaction or processing is allowed to continue.

In addition, you can add records to secondary files of Batch Update or Real-time Update programs. You can also delete records from secondary files of Batch Update or Real-time Update programs. Any secondary file can be accessed through any index which does not allow duplicates.

The variety of ways in which Data Manager allows you to specify multiple-file access in a single transaction constitute a very important feature of DM programming. The concept of primary-secondary file relationships is described more completely in the discussion of each DM program type later in this manual.

RECORD STRUCTURES

A data file created by B20 Data Manager must have a single record size. Record sizes vary from a minimum of 4 bytes to a maximum of 1017 bytes. The maximum number of fields in a record is 100. Two record types are recognized in a Data Manager application file. The record types are distinguished by key value only.

RECORD KEYS

A file must have at least one index and can have up to five indices. The key of index number one must have its first field at the start of the record, and any other fields in this key must immediately follow. In other words, the keys of index number one must begin in the first byte of the record and must be contiguous. The keys of the other indices can start at any position in the record. Each key to a Data Manager file can be composed of up to 16 fields. The length of each of the five keys must not exceed 64 bytes.

Keys can be sorted in ascending or descending order. Duplicate keys are allowed in any index except index number one.

Alphanumeric key fields should not contain special characters which have an ASCII representation less than 20H.

A key cannot contain fields of Standard Descriptions 26-28 (i.e., integer, single-precision real number, and double-precision real number).

NOTE

> Use signed numeric fields in a key with caution because a minus sign (-) will collate higher than a plus sign (+).

CONTROL RECORD

Each file created with Data Manager contains a control record in which any numeric key fields are zero-filled, and any alphanumeric key fields are space-filled. The rest of the record is filled with hexadecimal zeros. This record can be accessed with a Maintenance program or an Inquiry program which allows the application user to enter a key consisting of all zeros and/or all spaces.

If you want to prevent the application user from accessing this record with a Maintenance program or Inquiry program, specify at least one key field in the program as a mandatory-entry field. This is a field attribute you select when you create the program with DMCREATE. If you use the mandatory-entry attribute, the application user is prevented from entering a key consisting of nothing but spaces or zeros. Report and Batch Update programs automatically skip the control record.

FIELD DESCRIPTIONS

Numeric fields in a Data Manager record can be stored in four different ways, as listed below. Note that for types 2, 3, and 4, you must enter both a Standard Description as well as a full description of the field.

1. As 8-bit numerics with the sign, if specified, in a leading byte. The maximum numeric field can contain a signed 15-digit number. This field requires 16 bytes in the data record. Decimal points are implied and a numeric field may be defined with any number of decimal places from zero to the length of the field.

   Although a maximum of 15 digits is provided for, large numbers are only precise to 14 digits.

2. As integer fields which require two bytes in the data record. The maximum integer field can contain a signed, 4-digit number.

   Note that although this field type is termed "integer," it allows for the specification of a decimal in the display description. The decimal point is implied, as described for type 1 above.

3. As a single-precision real number which requires four bytes in the data record. The maximum single-precision real number can contain a signed, 5-digit number.

4. As a double-precision real number which requires eight bytes in the data record. The maximum double-precision real number can contain a signed 15-digit number.

   Although a maximum of 15 digits is provided for, large numbers are only precise to 14 digits.

Note that due to system software limitations, accuracy is not guaranteed beyond the lower precision when you assign a lower precision to a higher precision real number. For example, assigning an integer of .1234 to a double-precision real number with 15 decimal places might result in the number .123399995267391. This problem does not occur if the receiving field is declared as having the same full description. In this case, the result in the example would be rounded correctly to .1234.

Alphanumeric fields in a Data Manager record can be from 1 to 45 characters long. All data representation is in ASCII, which is standard for the B20.

Data fields in a record can be described by using a Standard Description entry or a full description entry. Standard Descriptions identify a variety of commonly used alpha and numeric fields with sizes across the range of field sizes supported by Data Manager. A Standard Description is simply a 1- or 2-digit number which is associated with a particular field description.

Standard Description Examples	Explanation
5	A 5-character alpha field.
10	A 30-character alpha field.
15	A signed 4-digit whole-number field.

Whenever you have the option of entering a Standard Description in DMCREATE, you are provided with a reference chart which explains the meaning of each Standard Description.

An alternative to the Standard Description is to enter a full description. This is almost as easy, and it allows you to specify any field size supported by Data Manager. The following are sample field descriptions:

Full Description Examples	Explanation
6	A 6-character alphanumeric field.
8.0	An unsigned 8-digit whole-number field.
+8.0	A signed 8-digit whole-number field.
4.2	An unsigned 6-digit field with two decimal places.

These field descriptions indicate type (alpha or numeric) by the presence of a decimal point in all numeric descriptions. The sign attribute in numeric descriptions is specified by the prefix S, a plus sign (+), or a minus sign (-). Whole-number places precede the decimal point. The number of decimal places is entered after the decimal point.

FILE MANAGEMENT

Data Manager offers several file management features which enable you to control the creation and deletion of application data files.

File maintenance programs have a file management feature which determines what the program does when its primary file is not found. Since a missing file sometimes can be the result of incorrect operational procedures, one option is to have DMRUN terminate the Maintenance program and display an error message.

If you add records to a secondary file of a Batch Update or Real-time Update program, the existence of the file is checked. If the file does not exist, DMRUN uses the file management feature in the record description program to determine if the file can be created.

To allow the application user to create a file when normal operations require it, you can select an option to create a special Maintenance program which only creates a primary file. Another Maintenance program can be provided to maintain but not create a file.

To allow an application user to delete data files without using the Delete or ISAM Delete System utilities, you can create a Batch Update program which only deletes the data file.

For more specific information on these techniques, refer to the sections on Maintenance and Batch Update programming in this manual.

MAINTENANCE PROGRAM

A Data Manager Maintenance program serves two purposes:

1. It can be used to add, change, or delete records from a file.

2. It is a record description for the file it maintains. This record description will be used by other programs which reference the file.

After creating a Maintenance program for a file, you do not need to describe the corresponding file records again. If you create another program that references the same file, an Inquiry program, for example, you need to enter only the Maintenance program name in the Inquiry program specifications. When your Inquiry program runs, DMRUN extracts the record-description information from the Maintenance program and uses it for the Inquiry program as well. Be sure that all programs sharing record descriptions are in the same Program Library.

Whenever you create a program which references a file created or maintained by another program, indicate the name of the program which maintains the file to DMCREATE. DMRUN will extract the record descriptions at run time.

In addition to Maintenance programs, Menu programs and Real-time Update programs can also create files and provide record descriptions. Although we refer only to Maintenance programs in the following pages, the remarks apply to all three program types.

PROGRAM MODIFICATION AND RECORD DESCRIPTION

When you modify a Maintenance program, the automatic record-description feature of Data Manager has four important implications which you must understand regarding:

1. Accessing files.
2. Changing key information of primary files.
3. Changing key information of secondary files.
4. Changing program specifications.

Accessing Files

If you create a Maintenance program, use it to create a file, and then modify the program, you may have trouble accessing the existing file. If you add or remove fields from the program, or if you change the INCLUDE IN RECORD field attribute, the record size is changed at the same time.

Also, if you change the number of keys, the existing file will no longer be compatible. To correct this problem, follow the instructions in the subsection below, Changing Key Information of Primary Files.

The B20, like many computers, will not open an existing file unless the record size in your program matches the record size of your file. You will see Error 71, "A" ENTRIES OPEN/CLOSE ERROR when you try to run the modified program.

Use the following procedure to copy an incompatible data file to a new one so as to make it compatible with the current Maintenance program in which the number of indices has been changed.

1. Select Batch Update program.
   Select the Batch Update program to update the old file.

2. Assign Maint. programs.
   Refer to the old Maintenance program as the record description program for the primary file. Next refer to the new maintenance program as the record description program for the secondary file.

3. Write update function.
   Write an update function to assign data from the old to the new record.

4. Run program.
   Run the Batch Update program to create the new file.

Changing Key Information of Primary Files

If you created a maintenance program, used it to create a file, and then modified the program by changing the number of indices, the existing file will no longer be compatible with this program.

If you have re-created the file using a Batch Update program as indicated above, the following steps are unnecessary because the control record has already been updated for you. Otherwise, you will need to follow the procedure below.

1. Reorganize file.
   Use Direct Program Selection in DMRUN. Enter 7 (ISAM reorganize) for Program Class, and enter the library name and the Maintenance program name.

2. Remove Mand. Entry attribute.
   Temporarily remove the Mandatory Entry attribute from all the key fields in the Maintenance program.

3. Run program & enter values.

   Run the Maintenance program. Enter a zero (for a numeric value) or blank (for an alpha value) in the primary key field, and enter zeros or blanks in the new key field(s). Next press GO. This step causes the control record to be corrected for the new key specifications.

   Note that you may need to remove additional editing attributes to facilitate entering the values in this step.

4. Replace necessary editing attributes.

   When the control record has been corrected, you should modify the program to replace the necessary editing attributes.

Changing Key Information of Secondary Files

If a Maintenance program uses secondary files, you may need to respecify key fields for these secondary files when you modify the program. If you change the primary-file record description, the secondary-file key offset and length may not be valid. To correct this condition, give your secondary-file key specifications to DMCREATE again.

Changing Program Specifications

The automatic record-description feature allows you to modify a Maintenance program and change a record description. You do not need to make those changes in any other programs. As stated above, DMRUN automatically begins using a new record description in all other programs that reference the same file.

There may be, of course, other specifications in the other programs that need to be changed. For example, in a report you may want to adjust the position of a column heading to allow for an output item with a different maximum size. In an Inquiry program that uses a Forms Editor form, you may want to modify the form to allow for a new field size. Programs using standard forms and Batch Update programs, however, may need no changes.

After creating and modifying a few Data Manager programs, these relationships become quite clear. You should, however, always be careful when changing record descriptions.

VARIABLE NAMES

Data Manager Variable Names are used to reference three kinds of variables:

1. Register Identifiers.
2. Field Identifiers.
3. Report Text Identifiers.

REGISTER IDENTIFIERS

Register Identifiers consist of a Register Letter and a 1- or 2-digit register number. Register Letters are P, X, or T. Examples of Register Identifiers are P16, X1, and T4.

P Registers

The P (Program) Registers are available to your application program for use in making calculations and for exchanging information between DM application programs. The following conventions are used with regard to Program Registers:

```
P1  thru P40 = Numeric Program Registers.
P41 thru P45 = Alphanumeric or Alpha Program Registers.
```

X Registers

The X Registers are special registers of the DMRUN program which are used to exchange information between DMRUN and your DM application program. For example, X-1 contains the system date. Further information on the X Registers can be found in Section 8 of this manual.

T Registers

The T (Total) Registers are used in conjunction with the automatic total and subtotal features of a DM report program. There are 16 registers for totals and subtotals. These are referenced as T1 through T16 or S1 through S16.

FIELD IDENTIFIERS

Field Identifiers begin with an A, B, C, D, E, F, G, H, or I. These letters are called File Letters and are associated with application data files accessed by your DM program. The letter A is always used to identify the primary file.

The second part of a Field Identifier is a Field Number. This is always a number from 1 to 100 that identifies a specific field in the data records of a file. Examples of Field Identifiers are:

A2 = The second field in the primary-file record.

B4 = The fourth field in a secondary-file record.

H11 = The eleventh field in the record of another secondary file.

SOURCE ATTRIBUTE

The term SOURCE is closely related to the Variable Names. A SOURCE is a field attribute you can enter with DMCREATE when creating any program types except Batch Updates and Standard Menus. In the interactive programs, if you enter a Variable Name as a SOURCE attribute for a field, DMRUN will display the contents of the variable in that field, assuming that you are either:

1. Adding a record.
2. Maintaining a record when the field is not included in the record.

In report programs, a SOURCE is an attribute of an output item. You can enter either a variable name or a Report Text Identifier for this attribute. DMRUN will then print the report text or the contents of the variable register when that output item is processed.

PROGRAM LINKAGE

Program Linkage is the main feature of Data Manager Menu programs, but it is available in all program types. This feature allows dynamic selection of the next program to be executed by DMRUN. If you use the linkage feature, a DM program can initiate:

1. Any DM program in the same Program Library.
2. Any ASSEMBLER, BASIC, COBOL, FORTRAN, or PASCAL program.
3. The B20 ISAM Reorganize Utility for any file defined by a record-description program in the same Program Library.
4. The B20 Submit Utility.

If you do not make any entries in the Link Information program section when creating a program with DMCREATE, the last preceding Menu program will be initiated when your program terminates.

If one or more programs are specified in the Link Information program section, the first program specified will always be initiated unless you set the X3 register. If X3 is set, the value in that register is the number of the program that will run next.

When a series of linked DM programs terminates, the last preceding Menu program will be initiated. To return to the last menu from a BASIC program, your program must contain an ERC% = EXIT statement to terminate the BASIC Editor. To return to the last menu after linking to the Submit utility, the last entry in the Submit file must be the command DMRUN. In all other cases the return to the last menu is fully automatic.

SECTION 3

INSTALLATION

The following section lists the software requirements of B20 Data Manager in both a development environment and an operational environment. See below for the software lists that suit your particular needs.

DATA MANAGER SOFTWARE REQUIRED FOR APPLICATION OPERATION

The following are B20 Data Manager files required for operation in an application environment:

1. DMRUN - Run module code file.
2. DMRDATA - Run module system file.
3. DMRFORMS - Run module system forms.
4. DMPLIST - Program List utility code file.
5. DMPDATA - Program List heading file.
6. DMINIT.SUB - System initialization Submit file. Not required on cluster systems.
7. DM.USER - Sign-on file.

OPTIONAL DATA MANAGER SOFTWARE FOR APPLICATION OPERATION

The two files listed below are optional B20 Data Manager software for application operation. Use them only if you want DMRUN initiated automatically when the B20 is powered up.

1. SYSINIT.JCL - This file must be copied to the <SYS> directory on a stand-alone workstation to provide automatic ISAM (Indexed Sequential Access Method) installation when the system is powered up. This file is required only if you intend to use the DMREXIT.USER sign-on file described below.

2. DMREXIT.USER - This file can be copied to the <SYS> directory. Its name can be changed to provide a selection on the system sign-on form. This ".USER" file will automatically initiate DMRUN and will make DMRUN the system ExitRunFile.

OPTIONAL SYSTEM SOFTWARE FOR APPLICATION OPERATION

The following file is optional system software that can be used for application operation.

ISAMReorganize.Run - This file can be copied to the <SYS> directory from the system-software release disk. Refer to the technical notes which accompany the system-software release to determine which release disk contains this file.

ADDITIONAL DATA MANAGER SOFTWARE REQUIRED FOR DEVELOPMENT

The following are additional B20 Data Manager files that are used in a developmental environment:

1. DMCREATE - Create module code file.
2. DMCDATA - Create module system file.
3. DMCFORMS - Create module programming and utilities forms.
4. DMCHELP - Create module help forms.

INSTALLATION ASSUMPTIONS

Before you install B20 Data Manager you should have done the following:

1. Installed system software and brought up the Executive's COMMAND prompt on the B20 screen.

2. You have completed the B20 Operations Training Course (EL6300), included with the system software and become familiar with the sections of the B20 System Software Operation Guide dealing with the B20 utilities, their operation, and the terminology used to describe them.

3. We suggest that you make at least one complete backup copy of your Data Manager release disks. You can use the Floppy Copy utility for this purpose.

RESULTS OF AUTOMATIC INSTALLATION

The results of the B20 automatic installation feature are listed below. You should become familiar with them before you begin to implement the instructions regarding automatic installation.

1. The Create Directory utility is used to make a DM directory on [SYS].

2. The New Command utility is used to create the following commands:

   a. DMCREATE to run <DM>DMCREATE
   b. DMRUN to run <DM>DMRUN
   c. DMPLIST to run <DM>DMPLIST
   d. DMUPGRADE2 to run <DM> DMUPGRADE2

3. On a stand-alone workstation, the Copy utility is used to copy the following files:

   a. [FO]<SYS>SDM.USER to [SYS]<SYS>DM.USER

```
:SignOnVolume:  SYS
:SignOnDirectory:  DM
:SignOnFilePrefix:
:SignOnPassword:
:SignOnExitFile:  [SYS]<SYS>EXEC.RUN
:SignOnChainFile:  [SYS]<SYS>SUBMIT.RUN
Submit
[SYS]<DM>DMINIT.SUB
:
```

b. [FO]<SYS>DMINIT.SUB to [SYS]<DM>DMINIT.SUB

This Submit file will automatically install ISAM with the recommended parameters for stand-alone workstations. (See the ISAM section below.)

4. On a master workstation, the Copy utility is used to copy [FO]<SYS>MDM.USER to [SYS]<SYS>DM.USER.

This ".USER" file is the same as SDM.USER, referenced above, except that there are no SignOnChainfile or parameters for the Submit utility.

5. The Copy utility is used to copy all files from [FO]<DM>* to [SYS]<DM>* from the Data Manager release disks.

INSTALLING ISAM PARAMETERS

One feature of B20 system software is the ISAM Install utility. Using this utility, you can set your own parameters for the Indexed Sequential Access Method (ISAM) files. Notice that files created under one set of parameters cannot be accessed under another, incompatible set. Therefore it is a wise procedure to establish one standard set of parameters to be used both in a developmental and an applications environment.

To help you standardize the ISAM parameters you will use on the B20, Burroughs has defined a set of suggested parameters suitable for stand-alone workstations. They are contained in the DMINIT.SUB file. These parameters will allow you to work with record sizes up to 500 bytes and with files having up to five indices.

On cluster master workstations, ISAM is installed automatically when the system is powered up. These ISAM parameters also allow you to work with record sizes up to 500 bytes and with files having up to five indices.

If you prefer to use other ISAM parameters than those recommended by Burroughs, you can modify DMINIT.SUB to suit your needs. However, you should be aware that your selection of ISAM parameters can have an effect on the amount of memory required for the ISAM subsystem. If you use the suggested ISAM parameters, there will be enough memory for maximum performance of Data Manager. If you use other parameters, the amount of memory available for Data Manager could be significantly less and could therefore affect the performance of your DM application programs.

AUTOMATIC INSTALLATION PROCEDURE

With the B20 screen displaying the Executive's COMMAND field, the automatic installation procedure is listed below:

1. Check the number of directories.

   Use the Volume Status utility, as described in the B20 System Software Operation Guide, to check the number of existing directories. Since it must be possible to create one more directory, the number of existing directories must be less than the maximum number you specified when you initialized the [SYS] volume.

2. Check the number of free file headers.

   Use the Volume Status utility to verify that there are at least 18 free file headers, or 16 free file headers if you do not install DMUPGRADE(n). These will be sufficient to install the system, but you will need several more for operation.

3. Check free pages of disk.

   Use the Volume Status utility to make sure that you have at least 2800 pages of free disk space needed to install the system, or 2500 pages of free disk space if you do not install DMUPGRADE(n).

4. Insert the first release disk.

   Insert the disk labeled DMR200A in the disk drive.

5. Type "SUBMIT".

   In the COMMAND field, type: SUBMIT.

6. Press RETURN.

7. Fill in Submit File field.

Type [FO]<SYS>INSTALL.SUB in the Submit File field.

8. Press GO.

The installation of Data Manager will begin.

9. Follow screen instructions.

Follow the instructions which appear on the workstation screen.

10. Delete unneeded files.

You can delete <DM>COPY.SUB using the Delete utility described in the B20 System Software Operation Guide, since this file is not required in the <DM> directory for normal operation of the system.

If you have program libraries which were created with level-1.0 or level-1.1 Data Manager, you can upgrade them to level 2.0 with the DMUPGRADE2 utility described in Appendix E of this manual.

You are now ready to use the Data Manager system.

SECTION 4

DMCREATE & DMRUN FUNCTION KEYS

B20 Data Manager has 18 function keys or key combinations which are used in DMCREATE and DMRUN. The function of some of these keys changes when used during DMCREATE or DMRUN.

DMCREATE FUNCTION KEYS

There are 16 DMCREATE function keys or key combinations. These keys are described below:

Key	Function
CANCEL	Void all entries on the current form without updating the program library.
CODE & Down Arrow	Terminate a field entry and move cursor down one line to the first column of that line.
CODE & Up Arrow	Move cursor up one line to the first column of that line. Data in the current field is not entered.
DELETE	Character delete.
Down Arrow	Enter the selected field and select the next field.
FINISH	Return to the previous form or to the Program Sections menu, or terminate DMCREATE. This key will not cause data to be entered in a specification. Available only on the first field on the form.
F1	Insert a field definition or a blank line in a table. Available only when lit.
F3	Advance to the next program section without entering any fields on the current form. Available only when lit.
F8	Print contents of screen. Available when lit.

F10	Delete an entire field definition or an entry in a table. This is the only way to delete field definitions, report output items, etc. Available only when lit.
GO	Enter any remaining fields on the form with the values that they contain and continue to the next form.
HELP	Display all available help forms. To exit a help form, press either CANCEL, Down Arrow, FINISH, GO, NEXT, or RETURN.
NEXT/RETURN	Enter the selected field and select the next field.
SHIFT & Down Arrow	Same as CODE & Down Arrow.
SHIFT & Up Arrow	Same as CODE & Up Arrow.
Up Arrow	Return to the previous field. Data in the current field is not entered.

DMRUN FUNCTION KEYS

There are 12 DMRUN function keys or key combinations. These keys are described below:

Key	Function
CANCEL	Void all entries on the current form without updating the data file.
CODE & Left Arrow	Move cursor to the first character in the field.
CODE & Right Arrow	Move cursor to the position after the last character in the field.
DELETE	Character delete.
Down Arrow	Enter the selected field and select the next field.
FINISH	Return to the previous menu. Terminate DMRUN.

F8 Print contents of screen. Available when lit.

F10 Delete the current primary file record. Available only when lit.

GO Enter any remaining fields on the form with the values that they contain and continue to the next form.

HELP Display all available help forms. To exit a help form, press either CANCEL, Down Arrow, FINISH, GO, NEXT, or RETURN.

NEXT/RETURN Enter the selected field and select the next field.

Up Arrow Return to the previous field.

SECTION 5

HOW TO CREATE AND TEST A DM PROGRAM

The following steps summarize the procedure for creating and testing Data Manager application programs. A detailed explanation of each step follows:

1. Create an application disk directory.
2. Initiate DMCREATE.
3. Create a Program Library.
4. Enter program specifications on DMCREATE programming forms.
5. Test your completed DM program.
6. Terminate DMCREATE.

CREATING AN APPLICATION DISK DIRECTORY

You can create a disk directory for your application programs and files. It is a good practice to always specify a disk directory in file IDs while using Data Manager. This will prevent accumulating many miscellaneous files in your DM directory. Ideally, the DM directory should only contain files from the Data Manager release disks.

Use the B20 Create Directory utility to create your application disk directory. Operating instructions for this utility are in the B20 System Software Operations Guide.

INITIATING DMCREATE

Follow the procedure below to initiate DMCREATE. Normally only the third step is required.

1. Reset workstation. If you have just completed installing the Data Manager release on a stand-alone workstation, ISAM cannot be installed. In order to ensure

that Data Manager is able to install ISAM with recommended parameters, press the reset button on the back of the lecturn to bootstrap the system.

This step is not necessary for a master workstation because ISAM is automatically installed with the correct parameters by the system software.

Normally this step will not be necessary if you use the sign-on name described in Step 2 each time you power up the system.

2. Enter DM, date, and time.

When the sign-on form appears, enter DM. Enter the date and time, if required.

This step is necessary only when you reset or power up the system.

3. Enter DMCREATE & press GO.

When the Command prompt appears, enter DMCREATE. (The abbreviation DMC normally is sufficient.) Then press GO. The Programming Activity Selection form of DMCREATE will appear within a few seconds.

Normally this is the only step that is required to initiate DMCREATE. You can simply use the DMCREATE command whenever you are certain that ISAM has been correctly installed.

CREATING A PROGRAM LIBRARY

Follow the procedure below to create a Program Library:

1. Select activity.

Select the CREATE/MAINTAIN PROGRAM LIBRARY activity on the DMCREATE Programming Activity Selection menu.

2. Enter file ID & press GO.

   Enter a file ID for your program library using any existing application disk directory name. Press the GO key.

3. Enter Library information (optional).

   Optionally enter the Program Library Control information. The significance of these entries is explained on the corresponding help forms.

4. Press FINISH twice.

   Press FINISH twice to return to the Programming Activity Selection menu.

ENTERING PROGRAM SPECIFICATIONS

Follow the procedure below to enter program specifications:

1. Select activity.

   Select CREATE/MAINTAIN DM PROGRAM on the DMCREATE Programming Activity Selection menu.

2. Enter file ID & program name.

   The Library and Program Information form will appear. Enter the file ID of any existing Program Library, using the appropriate disk directory name. Enter a program name.

3. Select program type.

   The Program Type Selection form will appear. Select one of the eight program types. Your first program should be a Maintenance program to create a file since most other program types require an existing file. Other entries on this form are not required.

   The Program Sections form for the type of program you selected will appear. The default selection is 1. Use this default value when you first create a program. When you are maintaining an existing program, this menu allows you to skip rapidly to any section of the program.

4. Enter specs.

Enter your specifications on the following forms as they are presented. Refer to the Entry Help and Form Help if you have a question about any of the entries.

5. Make changes, if needed.

If you need to go back and change a previous entry, press FINISH. This will return you to the Program Sections menu. From there, you can select the section of the program you want to change. You will skip to the beginning of the section you select. Page through the completed forms until you reach the one you want to change.

After entering your change, you can either press FINISH and return to the Program Sections menu again, or page through the programming forms and continue entering your specifications.

6. Complete entries & press FINISH.

When you have completed all the forms for the type of program you are creating, DMCREATE will redisplay the Program Sections menu. This allows you to skip to any section of the program and change previous entries. If you do not need to change any previous entries, press FINISH.

7. Create additional programs.

The Library and Program Information form will appear. If you want to create another program, enter a new Library File ID and/or Program Name. However, to test the program you have just completed, press FINISH.

TESTING COMPLETED DM PROGRAM

Follow the procedure below to test a completed DM program:

1. Select Test DM Program.

   When you have finished entering the program specifications, the DMCREATE Programming Activity Selection menu will appear. Enter selection 7, Test DM Program.

2. Choose direct selection.

   After a few seconds, the DMRUN Library Selection menu will appear. Enter selection 11, Direct Program Selection.

3. Enter file ID & program name.

   The DMRUN Direct Program Selection form will appear. The default entry for Selection Type is 1. Since this value is by far the most common, the field is automatically bypassed. If you wish to change the entry value, press the Up Arrow key and enter the desired value.

   Enter the Library File ID and Program Name for the program you just created. This Library File ID will become the default value.

   Your program will be initiated.

4. Complete or terminate.

   Depending on the type of program you created, it will either terminate automatically when processing is complete, or continue to run until you press FINISH.

   When your application program terminates, the DMRUN Library Selection menu will appear. You can either run another program by selecting 11 for Direct Program Selection, or terminate DMRUN by pressing FINISH.

TERMINATING DMCREATE

After you have tested your program, the DMCREATE Programming Activity Selection menu will appear within a few seconds. You could, at this point, create a new program, modify your existing program, return to the DMRUN for more testing, or terminate DMCREATE.

Do the following to terminate DMCREATE:

Press FINISH.	When the DMCREATE Programming Activity Selection menu is present, press FINISH.

SECTION 6

HOW TO INSTALL AND RUN APPLICATION SYSTEMS

The procedure for running Data Manager application systems is summarized in the five steps listed below:

1. Initiate DMRUN.
2. Install the Program Library. (optional)
3. Select the Program Library.
4. Select and Run the Application Program.
5. Terminate DMRUN.

A detailed explanation of these steps is presented in the following pages of this section.

INITIATING DMRUN

There are two methods of initiating DMRUN, as described below.

Method 1:

1. Sign on. Sign on with a selection of DM.

2. Enter DMRUN. Enter DMRUN as a command to the B20 Executive.

   If you use this method of initiating DMRUN, when DMRUN terminates, you will be returned to the B20 Executive and the Command prompt will appear.

Method 2:

1. Sign on. Sign on with a user name of DM.

2. Enter DMCREATE.

   Enter the command DMCREATE.

3. Select activity 7.

   When the Programming Activity menu appears, enter Activity 7, Test DM Program. This will initiate DMRUN.

   If you use this method of initiating DMRUN, you will be returned to the DMCREATE Programming Activity menu when DMRUN terminates.

INSTALLING A PROGRAM LIBRARY

This step is optional, since any DM program can be run by using the Direct Program Selection feature on DMRUN's Library Selection menu. While direct selection is suitable for testing individual programs, program selection for completed application systems is intended to be by menu only. To use the complete Data Manager menu feature, your Program Library must be installed in DMRUN according to the following instructions:

1. Select Utilities.

   When you initiate DMRUN, the Library Selection menu will appear. Enter Selection 12, Utilities.

2. Select Library Maintenance.

   The Data Manager Utilities menu will appear. Enter Selection 1, Library Maintenance.

3. Select Install.

   All currently installed Program Library descriptions will appear, along with all the Library Maintenance Options. Enter Option 1, Install a New Library.

4. Select Description & enter file ID.

   The Install Library Form will appear. Enter the Library Description and the Library File ID of the corresponding Program Library.

5. Press FINISH.

   The new Library Description will appear at the top of the screen under Currently Installed Libraries and the

Library Maintenance Options form returns. Press FINISH.

6. Press FINISH.

The Data Manager Utilities menu appears. Press FINISH.

The DMRUN Library Selection menu will appear, showing the description of the new Program Library.

7. Select Library.

If you have specified an Initial Program name in your Library Control Information for this library, and if that program has been created in the Program Library, you can now select your new library by entering its number as a selection on this menu.

SELECTING AND RUNNING AN APPLICATION PROGRAM

Follow the procedure below to select and run an application program:

1. Select Library.

To select an application program, first select a Program Library on the Library Selection menu. DMRUN will then run the Initial Program of that library.

2. Select program.

Select an application program on the Initial Program menu form. DMRUN will then run the application program.

The steps involved in running a Data Manager application program mostly depend on the features you have decided to include in the program.

3. Allow to run or press FINISH.

When a DM program or sequence of linked DM programs terminates, the last previous Menu program will be initiated. Depending on the type of Data Manager program involved, a program will terminate either when all the specified processing has been completed, or when you press the FINISH key.

4. Continue or press FINISH.

When the last previous menu appears, you can select another application on that menu, or press FINISH. If you press the FINISH key, the Initial Program menu form will be displayed.

If you press the FINISH key while you are on the the Initial Program menu, the DMRUN Library Selection menu will be displayed.

TERMINATING DMRUN

Do the following to terminate DMRUN:

Press FINISH.

DMRUN is terminated by pressing the FINISH key while the Program Library Selection menu is displayed. The result of the termination depends on how DMRUN was initiated. This is explained in the section on initiating DMRUN.

SECTION 7

FORMS INTERFACE

One of the important features of B20 Data Manager is that it enables the programmer to create different kinds of entry forms quickly and easily.

APPLICATION ENTRY FORMS

An Application Entry Form displays prompts and fields from an interactive DM program. The application user enters information in these fields under the control of the DM application program. The two types of application entry forms that B20 Data Manager offers are Standard forms and Forms Editor forms.

STANDARD FORMS

A Standard form is usually easier and more efficient to use than the Forms Editor form. Standard forms are created automatically by DMRUN from information you enter when you create a DM program. The form image is created each time your program is run. Therefore, even if you change your program, you do not need to make additional changes to the form. A new image will be created automatically from your modified program.

A Standard form can have a 48-character heading or title and up to 20 entry fields. The prompts for the entry fields are field names you assign while creating the program. They can be up to 16 characters long.

The form image has the field prompts in a column on the left side of the screen. Each entry field is left-justified in a column to the right of the prompts. Selected fields are highlighted in reverse video and unselected entry fields are not highlighted. Keyboard operation for Standard forms is identical to keyboard operation for Forms Editor forms. The form types can therefore be intermixed in a group of related DM programs.

FORMS EDITOR FORMS

Forms Editor forms offer almost complete flexibility in form layout and design. You can include explanatory information as well as field prompts on your forms. The positioning of text and entry fields is restricted only by screen size. Line drawing and a variety of highlighting options are also available. See the Forms Manual (form 1148715) for additional information on Forms Editor features and operating instructions.

If you decide to use a Forms Editor entry form for one of your DM programs, follow the procedures listed below:

1. Create form.

   Create the form first. A form name can be a maximum of 12 characters long. Field names can be a maximum of 16 characters long. The field names entered during program creation must be exactly the same as the field names entered during form creation.

   When you create the form, you can specify a default value, the Auto Exit option, the Display Default option, and any combination of highlighting attributes. Do not use the Repeating Field feature since it is not supported by this version of B20 Data Manager.

   Your form can use all but the bottom line on the screen. DMRUN uses this line for error messages.

2. Create Forms Library.

   When the form is completed, use the librarian utility to create a Forms Library containing the form. Notice that a ".form" suffix is added to your form name when it is saved by the Forms Editor. The suffix is removed again when you add the form to the Forms Library.

3. List form specs.

   Use the FREPORT utility to list the form specifications.

4. Enter Forms Library name.

   Use the Create/Maintain Program Library activity of DMCREATE to enter your Forms Library name in the control record of a Program Library.

   When you use DMCREATE to create a program, use the Program Library that references the Forms Library containing your forms.

5. Select Forms Editor.

   One of the programming forms DMCREATE displays is called the Form Selection menu. Specify the Forms Editor form option on this menu.

6. Enter form name.

   When DMCREATE displays the FORM NAME prompt, enter your form name exactly as it appears on the FREPORT listing of the Forms Editor form.

7. Enter field names.

   When you enter field definitions in DMCREATE, enter the field names exactly as they appear on the FREPORT listing.

8. Enter field specs.

   When entering Descriptions or Standard Descriptions on the DMCREATE Field Definitions form, the field lengths must agree with the field lengths of the FREPORT. An alphanumeric field on the form which is not used for entry can be shorter than the corresponding field in your program. DMRUN automatically truncates the field on output. Fields on the form which are not used for output can be shorter than the corresponding fields in DMRUN. Any other mismatching of field size in the form and the program will result in a run-time program error.

   The fields on the form are processed in the order you specify them in the program.

   DMRUN ignores any field on the form not referenced in the program.

If a field in the program has a blank field name or the field attribute Include on Form set to N (no), the field cannot trigger an interaction. However, the field can serve other purposes. See Section 10 for additional information on attribute usage.

APPLICATION HELP FORMS

All Data Manager application programs have an optional help form feature. Application help forms are created with the Forms Editor utility and are stored in a Forms Library. All of the help forms for programs in a particular Program Library must be stored in its corresponding Forms Library. This means that help forms can be shared by programs that process the same information. Help forms have a maximum size of 2000 bytes. This form size is reported by the Forms Editor.

The procedure for creating these forms is much the same as the procedure for creating entry forms, but the help form names are limited to four characters in length. There are three categories of help forms available in interactive DM programs:

1. Entry Help forms.
2. Program Help forms.
3. Error Help forms.

Reporting and Batch Update programs have only Error Help forms available. Help forms can contain only explanatory information. The application user is not permitted to make any entries on a help form.

The application user has access to Entry Help and Program Help forms from any field in an application program. When the cursor is in a field and the HELP key is pressed, the Entry Help form for that specific entry appears first. If the HELP key is pressed again, the Program Help form appears. If the HELP key is pressed while an error message is displayed, the appropriate Error Help form appears before the normal Entry and Program Help forms.

If you do not want to create your own Entry Help, Program Help, and Error Help forms, DMRUN has default forms which appear when the application user requests help.

ENTRY HELP FORM

The Entry Help feature gives a DM programmer the option of creating a help form for each entry in an application program. This help form might be used to explain what information to enter and how the information is used. It can also explain any validation that applies to the entry. Entry Help Form names are specified on the Field Definition form in DMCREATE.

PROGRAM HELP FORM

The Program Help form consists of a single help form which provides general information about the whole program. The name of this form is entered on the Program Type Selection form in DMCREATE. An explanation of what the program does, when it should be run, and any other features of the program which may not be entirely obvious might be included on a Program Help form.

ERROR HELP FORM

The Error Help forms are closely associated with DMRUN's error message conventions. The use of Error Help forms is described in the section of this manual on DMRUN Error Conditions.

DMRUN SYSTEM FORMS

All of the forms used by the DMRUN program are Forms Editor forms. They are supplied with the Data Manager product and are stored in the DMRUN Forms Library which is called DMRFORMS. This library contains both the System Entry forms and the default help forms used by DMRUN.

DMCREATE SYSTEM FORMS

All of the System Entry forms used by DMCREATE are in the System Forms Library called DMCFORMS. All the system help forms used by DMCREATE are in a separate library called DMCHELP. The DMCHELP Library is optional and can be kept off line after you become familiar with the features of DMCREATE.

The help forms in DMCREATE can be referenced by pressing the HELP key. The first form to appear gives you specific information about the field you are entering. On some forms, a second press of the HELP key provides additional, general information about the entire form that is displayed.

SECTION 8

DATA MANAGER FUNCTIONS

In all types of Data Manager programs except Standard Form Menus, you can choose sequences of function statements to specify intra-field entry validations, calculations for computed input and output items, record-selection criteria, and update processing. These function statements form a simple procedural language which is made up of only 12 constructs. Functions are completely optional in all but the update program types.

By using function statements you can easily include sophisticated features in your DM programs. You will use programming techniques with which you are familiar. This method is much faster than writing the entire program in a traditional programming language.

The functions included in DM programs are listed below by program type:

RUN-TIME DATA PROGRAMS

Initial Function: Called only once in each run of the program, before the application entry form is displayed.

Repeating Function: Called after all entries on the form have been completed, but before any files have been updated.

The entry form is presented only once in a Run-time Data program each time the program is executed. Therefore, the Repeating function gets called only once.

REAL-TIME UPDATE PROGRAMS

Initial Function: Called only once in each run of the program, before the application entry form is displayed.

Repeating Function: Called after all entries on the form have been completed, but before any files have been updated. It repeats each time a form is processed.

MENU PROGRAMS

Initial Function: Called only once in each run of the program, before the application entry form is displayed.

Repeating Function: Called after all entries on the form have been completed, but before any files have been updated.

The form is presented only once for each run of the program. Therefore, the Repeating function gets called only once.

MAINTENANCE PROGRAMS

Initial Function: Called only once in each run of the program, before the application entry form is displayed.

Repeating Function: Called after all entries on the form have been completed, but before the primary file has been updated. It repeats each time a form is processed.

INQUIRY PROGRAMS

Initial Function: Called only once in each run of the program, before the application entry form is displayed.

Inquiry Function: Called after all key fields on the form have been entered and all files have been read, but before any fields are displayed.

BATCH UPDATE PROGRAMS

Initial Function: Called only once in each run of the program, before any files have been read.

Update Function: Called once for each record in the primary file after the primary and all secondary files have been read, but before any files have been rewritten.

REPORT PROGRAMS

Control File Record Selection: Called immediately after a record is read from the Control file.

Primary File Record Selection: Called immediately after a record is read from the primary file.

Secondary File Record Selection: Called immediately after all secondary files have been read.

Report Header Function: Called immediately before the Report Header is printed.

Report Trailer Function: Called immediately before the Report Trailer is printed.

Page Header Function: Called immediately before the Page Header is printed.

Page Trailer Function: Called immediately before the Page Trailer is printed.

Continuation Page Header Function: Called immediately before the Continuation Page Header is printed.

Continuation Page Trailer Function: Called immediately before the Continuation Page Trailer is printed.

Detail Element Function: Called immediately before each Detail Element is printed.

VARIABLES

Several function statements can reference Data Manager variables. Here is an explanation of the different variables you can reference in the function statements.

P REGISTERS

The numeric Program registers P1, P2, P3...P40 are available for you to use in calculations and for other purposes. They all will hold a signed number with up to 10 whole number digits and five decimal places.

The P41, P42, P43, P44, and P45 registers are alphanumeric Program registers. They each hold up to 45 alphanumeric characters and can be used for any purpose.

X REGISTERS

The X Registers are controlled by the DMRUN program and have predefined usage. Read/write access to these registers is permitted unless otherwise noted below:

Register Function

X1 System Date. This register is restricted to read-only access.

X2 System Time. This register contains the time as a six-digit number in the format HH MM SS. Read-only access is permitted.

X3 Link Index. The value in this register is used by DMRUN as an index to your program linkage specification and is used to determine the next program to initiate. The default value is 1.

X4 Date input for the JULIAN, CJULIAN, and AGE function statements.

X5-X12 Input and output registers for the AGE function statement.

Registers X13 through X16 are maintained by DMRUN during report programs only.

Register Function

X13 Contains a count of detail elements in the level 1 group.

X14 Contains a count of detail elements in the level 2 group.

X15 Contains a count of detail elements in the level 3 group.

X16 Contains a count of detail elements in the entire report.

X17 If you specify a range of records, this register will contain the lower key. This register (and X18) is normally loaded automatically by specifying a range of records in the Run-time Data program. Registers X17 and X18 can also be loaded independently using the ASSIGNMENT statement. For more information on the range-of-records feature, see Section 11, Run-time Data program and Report program.

X18 If you specify a range of records, this register will contain the upper key.

X19 Range Flag. A value of 1 or T in this register tells DMRUN to process only a range of records in a Batch Update or Report program. The default value is 0, meaning process the entire file. More information on the range-of-records feature is provided in the section of this manual on Report Programs.

X20 Program Version. If you make a Program Version entry on the Program Type Selection form in DMCREATE, your entry will be loaded to this register when the program is initiated by DMRUN. This is a 16-character register. It is restricted to read-only access.

X21 Library Version. If you enter a Library Version ID entry while creating a Program Library with DMCREATE, your entry will be loaded to this register when the Library is accessed by DMRUN. This is also a 16-character register which is restricted to read-only access.

X22 Report Output Device. This register is used by DMRUN to override the output device specified in a Report program. This register can be set by a Run-time Data program, or by the report itself. The values for the different devices are listed below:

1. Parallel interface printer
2. Spooler
3. Screen
4. Available printer
5. Specific device/file

If you use the Available Printer option (value

4), the program will first try to print to the spooler. If the spooler is not installed, the program will automatically revert to direct printing.

When you use this option, be sure to select the Available Printer option only if a spooler is installed or a printer is connected to the B20. If you try to direct print a report from a station which does not have a printer installed, the report will halt. In this case, to continue you must either attach a printer or press ACTION/FINISH.

If you use the Specific Device/File option (value 5), the report can be directed to a specific device or a specific disk file. See register X23.

If X22 is not set by a function statement, the output device will remain as specified in the Report program.

X23 Device/File Name. If register X22 is set to value 5, you can specify a device or file name to which the report is to be written. If you specify a device, such as a line printer, you must place the device code in square brackets, e.g., [LPT]. If X23 is not set by a function statement, the device/file name will remain as specified in the Report program.

T REGISTERS

Total registers T1, T2, T3...T16 are used to accumulate totals during reporting. They can also be referenced by function statements in a report to calculate statistics. You should review the information in the Report Programming section of this manual before using these.

FIELD IDENTIFIERS

The variables mentioned above are called registers, indicating that they are all fixed-format internal storage areas. The next variables are called Field Identifiers because they identify disk-file data fields.

Examples of Field Identifiers are:

A1, A2, A3...A100 for primary-file fields.

B1, B2, B3...B100 for secondary-file fields.

Since the files accessed by a DM program are identified internally by File Identification Letters (A, B, C, D, E, F, G, H, and I), other secondary-file Field Identifiers could be created by using these letters along with a Field Number. C1, D12, E43, F19, G7, H4, and I100 would all be valid Field Identifiers.

Notice that by convention the primary file is always assigned the File Identification Letter A.

The Field Number part of each Field Identifier is a number assigned by DMCREATE when you create the Maintenance program for the file. You can obtain a cross reference of Field Names and Field Numbers by listing the Maintenance program using the program list feature in either DMRUN or DMCREATE.

FUNCTION STATEMENTS

There are 12 function statements which can be used in the Function sections of Data Manager programs. They are explained in the following pages.

1. AGE statement.
2. ASSIGNMENT statement.
3. CANCEL statement.
4. CJULIAN statement.
5. CONCATENATION statement.
6. DELETE statement.
7. END statement.
8. ERROR statement.
9. EXIT statement.
10. GOTO statement.
11. IF statement.
12. JULIAN statement.

AGE STATEMENT

The AGE statement can be used in all program types. The syntax of this statement is listed below:

```
——>AGE———> I
```

Be sure you have loaded the X4-X6 registers before executing this statement. These registers are described below:

Register	Function
X4	Date by which age period is to be calculated.
X5	Amount to be aged.
X6	Length of age period in days.

After the AGE statement is executed, the value in register X5 will be assigned to registers X7 through X12, depending on a comparison of the system date and the date in register X4.

Register	Function
X7	Future period.
X8	Current period.
X9	First past due period.
X10	Second past due period.
X11	Third past due period.
X12	Fourth past due period and older.

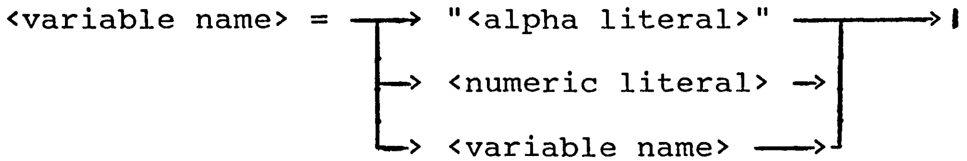
ASSIGNMENT STATEMENT

An ASSIGNMENT statement can be used in all program types, but do not apply to Total Registers in reports. You can assign any of the following to a variable name:

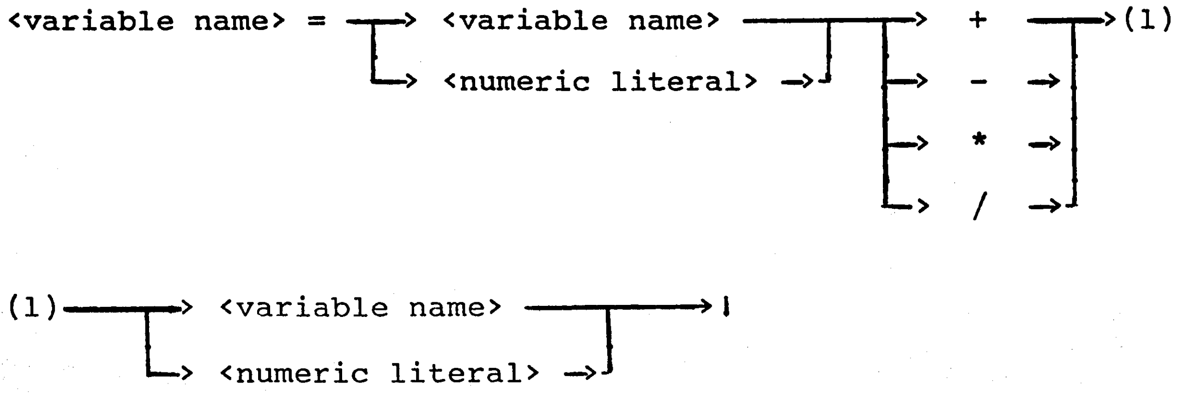
1. A numeric literal.
2. An alpha literal.
3. A variable name.
4. The result of an arithmetic operation performed on two variable names or a variable name and a numeric literal.
5. The result of a concatenation performed on two variable names or a variable name and an alpha literal.

The following represent the three correct forms of syntax for the ASSIGNMENT statement.

Variation 1:



Variation 2:



Variation 3:

```
<variable name> = ──┬──> <variable name> ──┬──>  &  ────>(1)
                    └──> <alpha literal> ──>┘

(1)────┬──> <variable name> ───┬────>|
       └──> <alpha literal> ──>┘
```

Examples:

```
P41 = "FRED"
A1  = 125
B2  = 1
A5  = A2*B3
P6  = P6 + 1
P41 = P42 & "FRED"
A3  = B3 & C5
```

CANCEL STATEMENT

The CANCEL statement can be used in all program types. It tells DMRUN to discontinue processing the last record(s) read and to proceed with whatever I-O comes next, depending on the program type. In a Report or Batch Update program, this statement is used to implement the record selection feature.

The syntax of this statement is listed below:

```
──>CANCEL───>|
```

CJULIAN STATEMENT

The CJULIAN statement can be used in all program types. Before it is executed, register X4 should contain a date in Gregorian notation. After the CJULIAN statement is executed, X4 will contain the original date expressed in century Julian notation.

The syntax of this statement is listed below:

```
     ——>CJULIAN———>|
```

CONCATENATION STATEMENT

See the ASSIGNMENT statement.

DELETE STATEMENT

The DELETE statement can be used only in Batch Update programs. It will cause DMRUN to delete the last record read from the specified file.

The syntax of this statement is listed below:

```
     ——>DELETE——> <file-identification letter> ———>|
```

Examples:

```
     DELETE A

     DELETE C
```

END STATEMENT

The END statement can be used in all program types. It causes DMRUN to terminate the program as if the application user had pressed the FINISH key. When you use an END statement in a Repeating function, all I-Os from the current transaction will be completed before the program terminates.

The syntax of this statement is listed below:

```
     ——>END———>|
```

ERROR STATEMENT

The ERROR statement can be used in any program except Report and Batch Update programs. It is used to indicate to the application user that the entries on the program form will not be processed. It prevents DMRUN from doing any file updates and causes one of five standard error messages to be displayed on the entry form.

The syntax of this statement is listed below:

```
    ——>ERROR——> <integer> ———>|
```

The integer must be from 1 to 5, inclusive.

Examples:

```
    ERROR 1
    ERROR 4
```

The text of the error message generated by the second example is:

```
    Entry validation error 4.
```

Refer to the section in this manual on DMRUN Error Conditions for more information on validation errors.

EXIT STATEMENT

The EXIT statement can be used in any program type. It causes DMRUN to stop executing function statements and to continue processing as though the end of the function statements had been reached.

The syntax of this statement is listed below:

```
    ——>EXIT———>|
```

GOTO STATEMENT

The GOTO statement can be used in all program types. It causes an unconditional branch to another function statement.

The syntax of this statement is listed below:

```
     ——>GOTO——> <line number> ———>|
```

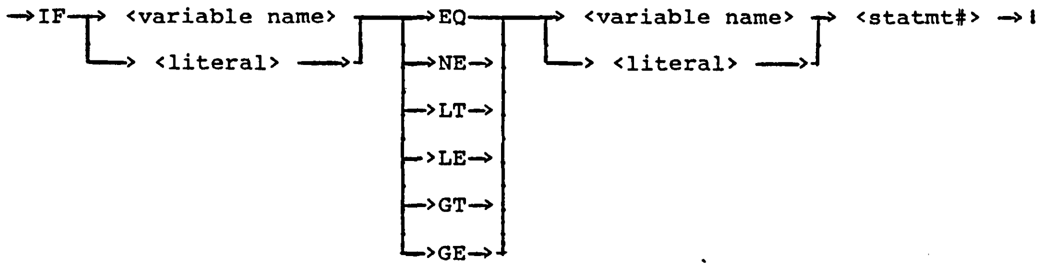
Example:

```
     GOTO 8
```

IF STATEMENT

The IF statement can be used in all program types. It is used as a conditional branch to another function statement. The condition is a comparison of two data items.

The syntax of this statement is listed below:



The abbreviations for the conditional operators are interpreted as follows:

Abbreviation	Meaning
EQ	Equals.
GE	Greater than or equal to.
GT	Greater than.
LE	Less than or equal to.
LT	Less than.
NE	Not equal to.

Examples:

```
IF A2 EQ P3 5
IF A3 NE B2 6
```

JULIAN STATEMENT

The JULIAN statement can be used in all program types. Before it is executed, register X4 should contain a date in Gregorian notation. After the statement is executed, the original date expressed in Julian notation will be in X4.

The syntax of this statement is listed below:

```
——>JULIAN———>|
```

SECTION 9

DMRUN ERROR CONDITIONS

The error conditions which DMRUN monitors are indicated to the application user by error messages. An error message is composed of a maximum of four parts. These parts appear in the order listed below. All error messages contain parts 1 and 3.

Part 1: A 3-digit error number at the beginning.

Part 2: Some error messages have a second part which indicates the file, field number, function statement, or report output item that caused the error.

Part 3: The error description.

Part 4: Some error messages have a 4-digit number after the error description. This is the operating system's error number which can be used to reference the B20 Operations Training Course (form EL 6300), Part II, Appendix A.

These error conditions have been divided into six categories according to the seriousness of the error and its probable cause. The categories are listed below:

1. Entry Validation Errors.
2. Intra-Field Validation Errors.
3. File Security Errors.
4. File Errors.
5. Program Errors.
6. System Errors.

You can minimize the impact of these errors on the application user in the manner described below.

ENTRY VALIDATION ERRORS

DMRUN automatically validates entries when data is entered in individual fields. Entry validation errors occur when an application user makes an entry which does not meet the requirements that were specified. These will be the most common errors encountered by the application user. The error messages for entry validation suggest corrective action but avoid technical terminology.

To help the application user avoid and/or resolve validation errors, you have the option of creating entry help forms for any of the entries in an interactive program. The help form for an entry is displayed when the cursor is in that field and the application user presses the HELP key.

You can also create a program help form which is available to the application user from any field in an interactive program. If you do not create field or program help forms, DMRUN will display a default help form when help is requested. The creation and specification of help forms is discussed in the Forms Interface section of this manual.

INTRA-FIELD VALIDATION ERRORS

In some interactive programs you may want to provide for intra-field validation. This validation is in addition to field-by-field validation which you specify during field definition. To provide intra-field validation, function statements can be specified in the Repeating function of your DM program.

The ERROR function statement allows you to display one of five standard error messages if the application user's entries do not satisfy your validation requirements. When the error message appears, the cursor will be in the first non-key field of the entry form in a Maintenance program, or the first field in other programs. Therefore, your explanation of intra-field validation requirements should be included in a field help form for the first field on your entry form.

You can include an explanation on the program help form (also available from the first field on the entry form).

See the Function Statements section of this manual for further information about the ERROR statement.

FILE SECURITY ERRORS

In a file Maintenance program, you can restrict the application user from adding, changing, or deleting records. If an application user attempts a restricted operation, DMRUN will display an error. To explain the restriction, you can create a security error help form. Enter the name of this form on the Primary Data File form in DMCREATE. The help form will be displayed by DMRUN if the HELP key is pressed while a security error message is on the screen.

If you do not specify a security error help form in DMCREATE, DMRUN will display a default help form when the application user presses the HELP key.

FILE ERRORS

Error messages in this category appear after an unsuccessful attempt to access a file or when abnormal invalid-key conditions occur. The corrective action for this category of errors varies from program to program. Therefore, DMRUN gives only general error messages which identify the file that is causing the error.

To clarify the error message and to suggest a specific corrective action, the application programmer can create an error help form for each file referenced by the program. This form is displayed if the application user presses the HELP key while a file error message is on the screen. If you do not create your own form, a default error help form is displayed when the application user presses the HELP key.

PROGRAM ERRORS

There is an additional category of easily understood error messages which relates to DM programming errors. These error messages indicate the specific cause of a problem in the DM program. The messages help you eliminate programming errors when you test your application programs.

To aid an application user who encounters a program error, you can create a program error help form. The name of this form should be entered on the Program Type Selection menu in DMCREATE. It will be displayed by DMRUN whenever the HELP key is pressed while a program error message is on the screen.

If you do not specify a program error help form in DMCREATE, DMRUN will display a default help form when the application user presses the HELP key.

SYSTEM ERRORS

Operational problems such as a failure of the hardware, system software, or Data Manager system can generate DMRUN error conditions. These are unlikely, but if one occurs, it will be identified by an error message.

SECTION 10

FIELD ATTRIBUTE COMBINATIONS

Many of the operational features of Data Manager programs are determined by your selection of field attributes on the Field Definition forms of DMCREATE. There are many possible combinations of attributes, but only a few of these normally are used. The most significant attributes are Include on Form, Allow Entry, and Include in Record. These have the greatest effect on the way DMRUN processes an entry.

INCLUDE ON FORM & FIELD NAME ATTRIBUTES

The Include on Form and the Field Name attributes determine whether or not DMRUN will attempt input or output to a form field. If Field Name is not blank and Include on Form is Y (yes), the field on the form can be used for input and/or output. If either Field Name is blank or Include on Form is N (no), DMRUN will not attempt any input or output to a form when the field is processed.

On Standard forms, the combination of entering a blank field name and setting Include on Form to Y has additional significance. This combination of attributes causes DMRUN to insert a blank line in the Standard form image where the prompt and entry area normally would appear. You can use this feature to separate groups of related entries.

ALLOW ENTRY ATTRIBUTE

When a field is included on a form, the Allow Entry attribute controls whether or not the cursor is positioned in the field to allow operator input. If you enter N (no) for the Allow Entry attribute, the field is bypassed during Add and Maintain activities and operator input is not permitted.

When adding records you must supply a source or automatic increment in order to store data by means of a field which does not allow entry. You can also store data in a no-entry field by specifying an assignment to the field in the Repeating function. When maintaining records, data in the records will be displayed but cannot be changed.

In both Add and Maintain activities, any specified validation is performed even if entry is not allowed. If there is a validation error, the application user must cancel the transaction, since it would be difficult, if not impossible, to correct invalid data in fields for which no entry is allowed.

The Allow Entry attribute is ignored when the Field Name attribute is blank or Include on Form is N (no).

INCLUDE IN RECORD ATTRIBUTE

The Include in Record attribute allows you to define fields which will either include or exclude data from storage in the data record. For example, when the Include in Record attribute is N (no) for a specified field, data in that field will not be stored in the data record defined by the program. Any information entered in a field that does not include data in the record is lost before the next field is processed, and is not available for reference by function statements. The most common use for this attribute is to specify fields which are used only to access or display information from secondary files for visual verification.

ATTRIBUTE COMBINATIONS

The different combinations of Include on Form, Include in Record, and Allow Entry which are supported by Data Manager are described in the Attribute Combinations Table below. These combinations are used to define the categories of all supported attribute combinations in the following tables. Attributes listed in these category tables can include Mandatory Entry, Source, Skip to Number, Field Duplication, and Automatic Increment.

Attribute Combinations Table

Category	Include on Form	Include in Record	Allow Entry
I	Y	Y	Y
II	Y	Y	N
III	Y	N	Y
IV	Y	N	N
V	N	Y	---

CATEGORY I ATTRIBUTE COMBINATIONS

Category I attribute combinations are the most common. In Add transactions, the application user can enter a field with Category I attributes and it will be stored in the data record by DMRUN. The field can be maintained in existing records.

Category I Table

	Mand. Entry	Source*	Skip To	Field Dup	Auto Inc
1.	N	N	N	N	N
2.	N	S	N	N	N
3.	N	R	N	N	N
4.	N	N	N	N	Y
5.	N	N	N	Y	N
6.	N	N	Y	N	N
7.	N	S	Y	N	N
8.	N	R	Y	N	N
9.	N	N	Y	N	Y
10.	N	N	Y	Y	N
11.	Y	N	N	N	N
12.	Y	S	N	N	N
13.	Y	R	N	N	N
14.	Y	N	N	N	Y
15.	Y	N	N	Y	N

*N No source, or source is primary-file field.
*S Source is secondary-file field.
*R Source is register.

CATEGORY II ATTRIBUTE COMBINATIONS

Category II Attributes provide operational features which can be useful in special circumstances. In Add transactions, data will appear in this field only if it is supplied from a source, field duplication, or automatic increment. The application user cannot change the supplied data. In Maintain transactions, the data appearing in this field will come from the primary file, but again, the application user may not change it.

Category II Table

	Mand. Entry	Source*	Skip To	Field Dup	Auto Inc
1.	N	N	N	N	N
2.	N	S	N	N	N
3.	N	R	N	N	N
4.	N	N	N	N	Y
5.	N	N	N	Y	N
6.	N	N	Y	N	N
7.	N	S	Y	N	N
8.	N	R	Y	N	N
9.	N	N	Y	N	Y
10.	N	N	Y	Y	N
11.	Y	S	N	N	N
12.	Y	R	N	N	N

*N No source, or source is primary-file field.
*S Source is secondary-file field.
*R Source is register.

CATEGORY III ATTRIBUTE COMBINATIONS

There are few occasions where Category III combinations are useful. You can use them to provide a skip to another field, or to provide a dummy non-key data field required by DMRUN in a Maintenance program when the actual record contains only key data.

Category III Table

	Mand. Entry	Source*	Skip To	Field Dup	Auto Inc
1.	N	N	Y	N	N
2.	N	N	N	N	N

*N No source, or source is primary-file field.
*S Source is secondary-file field.
*R Source is register.

CATEGORY IV ATTRIBUTE COMBINATIONS

Attribute combinations in this category normally are used to provide visual verification of information from secondary files. During both Add and Maintain transactions, the field will be initialized from its source. The cursor will skip the field to exclude entry of data by the operator. The contents of the field will not be stored in the data record.

Category IV Table

	Mand. Entry	Source*	Skip To	Field Dup	Auto Inc
1.	N	S	N	N	N
2.	N	R	N	N	N
3.	N	S	Y	N	N
4.	N	R	Y	N	N
5.	Y	S	N	N	N
6.	Y	R	N	N	N

*N No source, or source is primary-file field.
*S Source is secondary-file field.
*R Source is register.

CATEGORY V ATTRIBUTE COMBINATIONS

The most common reason for not including a field on a form is to create a record with a size greater than that which is required for the fields the application user will enter. This is done when you expect to define additional data fields in the record later. The additional fields are not included on the form, so application users will not be aware of these fields until they are actually used.

Note that while it is possible to provide a source and to specify validation for fields in Category V, this should be done with caution. Since all processing of these fields is invisible, any validation or existence-check errors that occurred would be difficult for the application user to correct.

Category V Table

	Mand. Entry	Source*	Skip To	Field Dup	Auto Inc
1.	N	N	N	N	N
2.	N	S	N	N	N
3.	N	R	N	N	N
4.	N	N	Y	N	N
5.	N	S	Y	N	N
6.	N	R	Y	N	N
7.	Y	S	N	N	N
8.	Y	R	N	N	N

*N No source, or source is primary-file field.
*S Source is secondary-file field.
*R Source is register.

COMPILATION OF TABLES

Some of the principles used in compiling the preceding tables may be helpful. These principles are listed below:

1. If a field name is not specified or Include on Form is set to N, the Allow Entry attribute is ignored.

2. Do not designate a Y (yes) for Mandatory Entry and a Skip to Number on the same field, as the skip will never occur.

3. Do not specify a field duplication or automatic increment on a field which has a Source attribute. The contents of the source variable will always replace the duplicated or incremented value.

4. A source which references a previously processed field in a primary record is valid. The contents of the first field will be used to initialize the second field. The possibility of a primary-file source is not included in the above attribute combinations.

SECTION 11

DATA MANAGER PROGRAM TYPES

If you have completed the programming exercises in the Data Manager Training Guide, you should have a general idea of how to create and use the various types of Data Manager programs. The preceding sections of this manual have presented general information about all the program types. This section has specific information on each of the eight types of programs.

MAINTENANCE PROGRAM

Use the Maintenance program to create and maintain files. This type of program offers a variety of features.

FILES

A Maintenance program can access a single primary file and up to eight secondary files. The primary file is the maintained file. Records in the primary file can be added, deleted, or changed. Secondary files can be referenced as sources for fields in the Maintenance program. These fields may or may not be included in the primary file depending on your selection of field attributes.

Up to five indices can be specified for any application file. A file can be maintained only through its primary index. DMCREATE will not allow an index number to be specify for a primary file in a Maintenance program.

If you change a record in a multiple-index file during maintenance, you can also alter any key in that record, before it is rewritten, except the key through which the record was read.

Note that an error will result if you attempt to add a record when a duplicate key exists in any secondary index which does not allow duplicates.

ATTRIBUTE COMBINATIONS

All of the attribute combinations presented in Section 10 are valid for all fields in a Maintenance program. At least one field which is a primary key to the primary file and one field which is not a primary key to the primary file must be entered from the keyboard when a Maintenance program is run. These fields do not need to be included in the record.

ADDITIONAL FEATURES

It is sometimes desirable to have special master-file Maintenance programs which only create the primary file, then terminate automatically. This is useful when your normal master-file Maintenance programs do not use the automatic file-creation option. (See DMCREATE form 093 - File Management in Appendix A.)

To make a master-file create program, do the following:

1. Make a copy of your normal Maintenance program.
2. Specify the automatic file-creation option for the copied version.
3. Enter the END function statement as the first statement in the Initial function of the copied version.

You may need to have fields defined differently in the control record of a file than in the data records. If this is the case, the Maintenance program that creates the file should define the control record fields. You can make a separate Maintenance program which does not create the file to access data records.

In some applications you may need to access data for updating or reporting purposes using field definitions that are different than those used for maintenance. Data Manager allows you to create multiple Maintenance programs for the same file. Use one version of the Maintenance program for actual maintenance of the file. Create another Maintenance program with revised field descriptions. Reference the second Maintenance program as the record-description program in your update or report.

PROCESSING CYCLE

The following steps summarize form processing in a Maintenance program. These steps are repeated until the Maintenance program is terminated.

1. Initialize form.

2. Enter key fields. Build secondary-file keys, read secondary files as required.

3. Attempt to read primary file.

   If a record is found: display all data, proceed with the maintain transaction.

   If a record is not found: initialize all fields, proceed with "add" transaction.

4. Enter other fields. Build secondary-file keys, read secondary files as required.

   Cursor wraps around through non-key fields until the GO key is pressed.

5. Process all fields in the program starting with the selected field.

6. Process Repeating Function.

7. Write/Rewrite primary file.

MAINTENANCE PROGRAMMING FORM SEQUENCE

The following sequence of forms appears when you create a Maintenance program:

Form Number	Form Name
070	Program Type Selection
080	Maintenance Program Sections
090	Primary Data File
093	File Management
100	Secondary Files Record Description Programs
110	Form Selection
120	Forms Editor Form*
130	Standard Form - Form Heading*
140	Numeric Range Table
145	Alphanumeric Range Table
150	Field Definition
550	Primary File Indices
169	Program Fields Used for Secondary File Key**
170	Initial Function
190	Repeating Function
200	Program/Utility Linkage
250	Application File Error Help

*Optional depending on Form Selection.
**Optional depending on Secondary Files Record Description Programs.

BATCH UPDATE PROGRAM

With a Batch Update program, you can update, create, or add records to as many as eight files, and update a ninth file (the primary file).

FILES

A Batch Update program can access a single primary and up to eight secondary files. The primary file is the file containing the transactions and is read sequentially in key order. You can access this file through any index. The index used to access the primary file allows duplicate keys. Secondary files are read randomly according to key value supplied in the primary-file transaction records. A secondary file in a Batch Update program can be used in one of four ways to:

1. Provide input from existing records for the transaction (read only).
2. Update existing records.
3. Add new records to an existing file.
4. Create a new file.

A Batch Update program can also be used to delete records from any of the files being accessed. This is done by using the DELETE statement in the update function. See the section on function statements for more information about DELETE.

You can access your secondary files through any index that does not allow duplicate keys.

If you change a record in a multiple-index file during an update, you can also alter any key in that record, before it is rewritten, except the key through which the record was read. When you add or change records, all keys are automatically updated.

Since duplicate keys are not permitted in the primary index of any file, any attempt to add a record when a duplicate key exists in the primary index will result in an error.

A similar error will result if you attempt to add a record when a duplicate key exists in any secondary index which does not allow duplicates.

ATTRIBUTE COMBINATIONS

Since there is no field definition feature in Batch Update programs, none of the attribute combinations are directly relevant to this program type.

ADDITIONAL FEATURES

A Batch Update program can be used to delete Data Manager application data files without first performing an update. If, for example, you have a transaction file which must be deleted daily, you can create a Batch Update program which does nothing else. Specify the record-description program for the file you want to delete on DMCREATE form 225 - Primary File Record Description Program. On the same form, indicate that you want the primary file deleted at the end of the update. Enter an END statement as the first function statement of the initial function of your program. The Batch Update deletes the file and terminates automatically.

Record selection is implemented in a Batch Update program by using the CANCEL function statement conditionally in the Repeating function.

A Range of Records update feature is available by entering starting and ending key values in a Run-time Data program preceding the Batch Update. The Run-time Data program should also assign the value T (true) to register X19.

Note that the lower and upper keys are stored in X Registers 17 and 18, respectively. These registers can be loaded directly by making assignments to them in the initial function of this program. Using this method requires that X19 be set to T (true). This method is described more fully under Run-time Data program.

PROCESSING CYCLE

The following steps summarize the update processing in a Batch Update program. These steps are repeated until:

1. Read primary file.
2. Read all secondary files.
3. Process the update function.
4. Add new records.
5. Rewrite all updated records.

BATCH UPDATE PROGRAMMING FORM SEQUENCE

The following sequence of forms appears when you create a Batch Update program:

Form Number	Form Name
070	Program Type Selection
210	Batch Update Program Sections
225	Primary File Record Description Program
103	Secondary Files Record Description Programs
160	Primary File Fields Used for Secondary File Key*
170	Initial Function
195	Update Function
200	Program/Utility Linkage
250	Application File Error Help

*Optional depending on Secondary Files Record Description Programs.

REAL-TIME UPDATE PROGRAM

You can use the Real-time Update program to update, create, or add records to files. If you want an ongoing record of transactions (i.e., an audit file), use the Transaction Log.

FILES

A Real-time Update program can have up to eight secondary files. The secondary files are accessed randomly according to key values entered as a part of the transaction. A secondary file in a Real-time Update program can be used in one of four ways to:

1. Provide input from existing records for the transaction (read only).
2. Update existing records.
3. Add new records to an existing file.
4. Create a new file.

The primary file in a Real-time Update program is optional and is called a Transaction Log. This file can contain transaction data, secondary-file data, and computed items.

An application file can be accessed through any one of its indices that does not allow duplicate keys.

If you change a record in a multiple-index file during an update, you can also alter any key in that record, before it is rewritten, except the key through which the record was read. When you add or change records, all keys are automatically updated.

Note that an error will result if you attempt to add a record when a duplicate key exists in any secondary index which does not allow duplicates.

FIELD ATTRIBUTE COMBINATIONS

Any attribute combinations which result in duplicate Transaction-Log keys are invalid in a Real-time Update program. All other combinations listed in Section 10 are valid. At least one field must be entered from the keyboard when a Real-time Update program is run.

ADDITIONAL FEATURES

Since Transaction-Log records can be defined by Real-time Update programs, these programs can be referenced as record descriptions by other Data Manager programs in your application system. A common example would be to reference a Real-time Update program as a record description for a report. This is the easiest way to create a report which prints a transaction journal from information in a Transaction-Log file.

TRANSACTION-LOG FILES

The Transaction Log in a Real-time Update program is referred to as an "add-only" file. It is called an add-only file because records in a log file can be added but not maintained. Since duplicate keys are not allowed in add transactions, each transaction to be logged must contain a unique key. Unless the primary-key fields contain a unique value, DMRUN does not process a Real-time Update transaction when the Transaction Log is being used.

Creating Key Fields

The way in which you provide unique keys for Transaction-Log records depends on how the Transaction Log is used in your application system. The easiest way to ensure that log records contain unique keys is to define an automatic increment field as the primary-key field and to prohibit the application user from entering this field. The key field then becomes a simple transaction number which increases with each transaction processed.

To use this method, you must have an empty Transaction Log file each time the Real-time Update program is initiated. You need an empty Transaction-Log file because the automatic increment field is initialized to the amount of the increment each time the program is run. Therefore, the first log record of the first run has the same key as the first log record of any other run of the program.

However, if your system provides some means of copying or deleting the log file at the end of each run of the update program, DMRUN will automatically creates a new, empty log file at the start of the next run.

When you use Transaction-Log files, remember that there is only one file name for the Transaction Log specified in the program. Therefore, on a cluster system, each application user running the same Real-time Update program is adding records to the same log file. If all programs use a simple transaction-number key as described above, concurrent use of the programs results in duplicate-key errors.

Duplicate-key Error Resolution

There are four methods of resolving this problem of duplicate-key errors. These methods are listed below:

Method 1:

If you must have the transactions in separate files, create copies of the program and change the file name of the log file in each copy. Each workstation should use a separate copy of the program.

Method 2:

Another way to provide separate log files for each workstation is to create one disk directory for files that are shared by all workstations and a separate directory for each workstation which must reference non-shared files. Specify a directory name in the file ID for shared files. Do not specify a directory name in file IDs for non-shared files. Then have each workstation logged on to its directory for non-shared files.

Method 3:

If you require a consolidated log file for all workstations in a cluster, you can create unique keys for this file by including an operator ID, workstation ID, or batch number in the log file key. The most convenient way to do this is to precede the Real-time Update program with a Run-time Data program. The Run-time Data program can prompt the application user to enter the workstation-dependent part of the key. That entry can be passed to the Real-time Update program by storing it in a program register. The program register is then used as a source for the first primary-key field. The transaction number can be used for a second primary-key field.

Method 4:

In some cases, unique keys for log files can be created by using registers X1 and X2 to supply a date and time stamp for each transaction.

Log File Access

While a Real-time Update program cannot reference existing records in a Transaction-Log file, the log file is accessible to other types of programs. You could, for example, create an Inquiry or Report program to provide information on previously entered transactions.

PROCESSING CYCLE

The following steps summarize form processing in a Real-time Update program. These steps are repeated until the program is terminated.

1. Initialize Form.

2. Get key fields (only if Transaction Log is specified).

3. Enter other fields, build secondary keys, and read secondary files as required.

Cursor wraps around through non-key fields until GO is pressed.

4. Process all fields in the program starting with the selected field.

5. Process Repeating Function.

6. Validate unique Transaction-Log key (only if Transaction Log is specified).

7. Write transaction log (only if Transaction Log is specified).

8. Add new secondary records.

NOTE

> If an error occurs when adding a record to a secondary file (e.g., a disk-full condition), all newly added records are deleted and the transaction is cancelled.

9. Rewrite all updated secondary records.

REAL-TIME UPDATE PROGRAMMING FORM SEQUENCE

The following sequence of forms appears when you create a Real-time Update program:

Form Number	Form Name
070	Program Type Selection
082	Real-time Update Program Sections
097	Transaction Log
103	Secondary Files Record Description Programs
110	Form Selection
120	Forms Editor Form*
130	Standard Form - Form Heading*
140	Numeric Range Table
145	Alphanumeric Range Table
150	Field Definition
550	Primary File Indices**
169	Program Fields Used for Secondary File Key***
170	Initial Function
190	Repeating Function
200	Program/Utility Linkage
250	Application File Error Help

*Optional depending on Form Selection.
**Optional depending on Transaction Log.
***Optional depending on Secondary Files Record Description Programs.

INQUIRY PROGRAMS

You can use an Inquiry program to display data on the screen.

FILES

An Inquiry program can access a single primary file and up to eight secondary files. You can access the primary file through any one of its indices. The index used to access the primary file allows duplicate keys. You can access secondary files through any index that does not allow duplicate keys.

An Inquiry program can supply information from no more than one record in each file accessed. Whenever information from more than one record per file is required, you should create a Report program with output to the screen.

ATTRIBUTE COMBINATIONS

All field-attribute combinations are valid for key fields in an Inquiry program. The only attributes which can be applied to non-key fields are Field Name and Source. A Source attribute must be supplied for all fields in the Inquiry program. In addition, a Description or Standard Description must be supplied whenever the source is a numeric Program Register. All other attributes applied to non-key fields are ignored.

At least one key field must be entered from the keyboard when an Inquiry program is run. This field does not need to be included in the record.

ADDITIONAL FEATURES

Computed items can be included in an Inquiry program by specifying calculations in the Inquiry function, storing the results in program registers, and then declaring those registers as field sources in the subsequent non-key field.

As indicated below, the Inquiry function is processed before any data is displayed. This means that the results of any calculations are available, along with information from the files.

Calculated inquiry fields reference a numeric Program Register as a source. Whenever you use a numeric Program Register as a source, you must also specify a Description or Standard Description. If the number of whole number places specified in the description is insufficient to display all significant whole number digits in the Program Register, the application program will be terminated. If the number of decimal places specified in the description is insufficient to display all significant decimal digits in the Program Register, the contents of the Program Register will be rounded and displayed.

PROCESSING CYCLE

The following steps summarize form processing in an Inquiry program. These steps are repeated until the program is terminated.

1. Initialize form.
2. Enter key fields.
3. Read primary file.
4. Read all secondary files.
5. Process Inquiry function.
6. Display all data.

INQUIRY PROGRAMMING FORM SEQUENCE

The following sequence of forms appears when you create an Inquiry program:

Form Number	Form Name
070	Program Type Selection
240	Inquiry Program Sections
220	Primary File Record Description Program
100	Secondary Files Record Description Programs
110	Form Selection
120	Forms Editor Form*
130	Standard Form - Form Heading*
140	Numeric Range Table
145	Alphanumeric Range Table
155	Field Definition
160	Primary File Fields Used for Secondary File Key**
170	Initial Function
180	Inquiry Function
200	Program/Utility Linkage
255	Application File Error Help

*Optional depending on Form Selection.
**Optional depending on Secondary Files Record Description Programs.

REPORT PROGRAM

FILES

A Report program can reference a primary file and eight secondary files. The primary file can be accessed through any one of its indices. Records in the primary file are always accessed index-sequentially.

Secondary files can be accessed through any index which does not allow duplicates. Secondary-file records are accessed randomly based on a key value obtained from each primary-file record.

One of the nine files is identified as the control file. The control file is usually the primary file. Its significance is explained in a separate section below.

In a report that only accesses one file, you should identify that file as the primary file. In a report that accesses more than one file, you should select the primary file by considering which file you want to access index-sequentially.

For example, suppose you want to create a report of transactions from a transaction file and include information from a customer master file with each transaction. The key to the customer file is customer number. The key to the transaction file is customer number and transaction number. The transaction file key contains two fields because there can be several transactions in the file for a single customer.

If you use the customer master file as the primary file, you will read the customer file index-sequentially. When you read the first customer record from the customer file, you will know the Customer Number associated with this record. That, however, is insufficient to randomly access all of the possible transactions for that customer in the transaction file. The problem, of course, is that the customer records do not supply us with transaction numbers to use as keys.

If you use the transaction file as the primary file you will read the transactions index-sequentially. When you read the first transaction record, you will be able to access the customer record randomly. Therefore, the customer file is a secondary file. Once you have all the data from both files, you can print output items for the first transaction and read the next transaction file record. In this manner, all of the transaction records will be processed. For each transaction, you can obtain the customer information by randomly accessing the customer file. An actual read of the customer file will only occur when a transaction record references a customer other than the one in the previous transaction.

In the special case where two files have identical keys and a one-to-one correspondence of records, either file can be the primary file. If there is not a one-to-one correspondence of records, the primary file is again the one which you want to process sequentially.

REPORT TEXT

Report programs have access to the Report Text table. This table is used to store constant alphanumeric information for report titles, column headings, and miscellaneous other uses. The text is entered into the table on DMCREATE form 300 REPORT TEXT. The Text table entries can be referenced using a Report text identifier as a source for report output items. Report text identifiers consist of the letter R followed by an integer from 1 to 40. Examples of report text identifiers are R1, R15, R30.

LINE ADVANCES

The specification of a line advance serves as the signal to print the line. If data in an element is not terminated with a line advance, then no data will be printed after the previous line advance or the entire element (if there was no previous line advance).

DETAIL ELEMENTS

A Detail Element in its simplest form is one line of output from one primary-file record. In many cases, the data you wish to print from a primary-file record will not all fit on one printed line of the report. If you use two or more lines to print the information from the primary-file record, this group of lines is still considered a single Detail Element.

In other cases, you can access one or more secondary files in addition to the primary file to obtain report data. If data from the secondary files is printed along with the data from each primary-file record on one or more lines, this group of lines is also a Detail Element.

A more abstract example is provided by a summary type report. In a summary report the Detail Element is only used to specify the accumulation of totals or other computations. There is no actual printing. Only processing is specified.

We can give a definition of a Detail Element based on these examples. A Detail Element in a Data Manager report specifies the processing of one or more fields which come from or are associated with a single record in the primary file.

GROUPING

The simplest report structure contains only Detail Elements. If the records in a file are in the proper sequence, it is often desirable to group Detail Elements and provide group headers and/or trailers. Consolidating repetitious output items in headers and trailers produces a more concise report and allows the specifications of group totals or other statistics.

In the Report Body specifications of a Data Manager Report program, six different headers and trailers can be defined. These are all called Grouping Elements because they are used to create groups of Detail Elements. They are called Level 1, 2, and 3 Headers and Level 1, 2, and 3 Trailers.

A level-1 group is composed of one or more Detail Elements with either a level-1 header, a level-1 trailer, or both. A level-2 group is composed of one or more level-1 groups with either a level-2 header, a level-2 trailer, or both. As you might expect, a level-3 group is composed of one or more level-2 groups with either a level-3 header, a level-3 trailer, or both. What you might not expect is that within these Grouping Elements you can reference data from primary- and secondary-file records. You will always get data appropriate to the Element where the reference was made. In the case of group headers, file data will come from records associated with the next Detail Element to be printed. Group trailers get file data from records associated with the last printed Detail Element.

The dependence of all Grouping Elements upon Detail Elements has one other important consequence. No Grouping Elements are printed unless there is at least one Detail Element around which all the headers and trailers can be printed. If this is not convenient, see the section on control files below. By using the Control File feature, you can have a group header and/or trailer without any Detail Elements in the group.

An important point to keep in mind when designing Data Manager reports is that there must be a field in the primary-file records which defines group boundaries. This will normally be a key field or part of a key, although any single data field can be used. If a group boundary were defined by a change in two or more fields, you would have to use function statements to set a single Program Register based on an evaluation of the data fields. The Program Register could then be used to define the group boundaries.

PAGE SIZE

A single report element cannot exceed the physical page size if you use page headers and/or page trailers. In cases where you have an exceptionally large report element, it may be possible to adjust the physical page size to a value other than the actual page size and obtain the desired results.

SUMMARY REPORTS

In some reports, it may be desirable to print only intermediate and/or final totals and statistics, without all of the supporting Detail Elements. To create this type of report (summary report), do the following:

1. Specify a Detail Element which includes all the fields to be processed.
2. Specify total-register numbers for all fields to be totaled and any additional calculations in the Detail function.

NOTE

> Do not specify line advance or print position for any of the Detail Element output items.

To provide intermediate totals and statistics you should define Grouping Elements just as you would for a normal report. Grand totals and final statistics can be printed with a report trailer.

REPORT TOTALS

The Automatic Total feature of Data Manager reports can be used to create both totals and subtotals. The numeric values for both totals and subtotals are accumulated in 16 Total Registers. Although we refer to the registers as "Total Registers," the same registers are used for both totals and subtotals.

In order to use one of the Total Registers, specify the register number as an attribute of the output item for which you want a total. If a total is to reflect data from several output items, the same register number can be specified for each one.

Each of the 16 Total Registers can be used to accumulate up to three levels of intermediate totals as well as a grand total for the entire report. This means that DMRUN maintains four totals for each Total Register. Each register contains totals intended for reference in a level-1, level-2, and level-3 group element, and a total for reference in the report trailer.

When you reference register T1 in a level-2 trailer, for example, the level-1 total and level-2 total will be cleared. The level-3 total will continue to accumulate until T1 is referenced in a level-3 group element. The grand total will continue to accumulate until T1 is referenced in a report trailer. In other words, you can reference the same Total Register at different places in the same report, and DMRUN will automatically provide an appropriate value for each place.

The distinctions between totals and subtotals becomes important only when a Total Register is referenced as a source for an output item or in a function statement. For example, if Total Register 1 is referenced as T1, DMRUN assumes you want a total. This means that immediately after processing the current value, the register will be cleared as described above. If Total Register 1 is referenced as S1, DMRUN assumes you want a subtotal and the register is not cleared.

SIGN POSITION

You can specify a leading or trailing sign for all signed numeric output items. If you want all signed numeric fields in your report to be printed with trailing signs, then on form 290 - OUTPUT SPECIFICATIONS enter Y (yes) for the TRAILING SIGNS? prompt. The default value for this options is N (no). The default causes all signed numeric fields to be printed with leading signs.

DATE FORMAT

Any numeric field can be printed as a date, i.e., six digits without suppression of leading zeros. This is done by specifying a Numeric Edit value of 6 (print as date without separators) or 7 (print as date with separators) when you specify your Report Output Items.

REPORT OUTPUT TO THE SCREEN

One of the report output devices is the screen. The screen on a B21 will display a maximum of 80 characters per line. Reports intended for use on this workstation should be formatted so that report lines require 80 characters or less. Report lines with print positions past 80 will be wrapped around on the following line. Unless special care is taken when the report layout is designed, output to the screen with wraparound is less readable than a full-width printed version.

Since B22 workstation screens will display a 132-character line, the same reports can be output to the printer or to the screen without wraparound. On both workstation models you may want to specify a special physical page length for reports output to the screen. The default value of 66 is not suitable if you would like to have page headings visable on each complete screen full of report output.

An additional feature available on reports output to the screen is the option of obtaining a printed copy of the report image. The F8 key is enabled at the end of each screen of output and can be used to initiate printing.

PRINTER INTERFACE

Data Manager Report programs can interface to the B20 Spooler utility. On cluster systems, the spooler is required on the master workstation to provide shared access to the printers. If you are creating a report which is to be run on a workstation in a cluster, you will normally select spooler as the output device. DMRUN will then direct report output to the spooler and the print job will be queued behind any other print jobs currently waiting. If a Report program which calls for output to the spooler is run on a system when the spooler is not installed, the application user is given the option of directing the report directly to a parallel interface printer attached to the user's workstation.

A Report program with output to the printer can be run on any workstation that has a local parallel interface printer connected to it. The report will process normally, assuming a spooler is not installed. If a spooler is installed, the direct-print report is assumed to be high-priority output and will be printed on the local printer as soon as any currently printing spooled output is complete. After DMRUN releases control of the printer, the printing of jobs directed to the spooler can resume.

An attempt to do direct printing on a workstation without the provision for a local printer will result in an error message and an option to spool the report output.

You can also choose the Available Printer option. This causes the program first to try to print to the spooler. If the spooler is not installed, the program will automatically revert to direct printing.

If you use the Specific Device/File option, the report can be directed to a specific device or a specific disk file.

When using the F8 key to obtain a printed copy of a screen image, DMRUN attempts to send output to a spooler if available and, if not, directly to a local printer.

You should not allow an application user to send printed output directly to a printer unless a printer is actually available. System-software limitations prevent DMRUN from detecting this condition and providing an error message. In this case, the DMRUN program would have to be terminated using the ACTION-FINISH key.

The maximum length of a print line is 250 characters. However, be sure to set the length of the print line in your program according to the limitations of the printer. If the line is longer than the printer capacity, the line will still be printed, but the overflow will be printed on the next line (i.e., be wrapped around). This will cause the line count maintained by the system to be corrupted.

PRE-PRINTED FORMS

Data Manager Report programs can be used to produce invoices, statements, and a variety of documents which use preprinted forms. Since an invoice is one of the more complicated documents commonly required in application systems, it makes a good example of several Data Manager reporting features.

The information on a typical invoice can be divided into four categories:

1. Heading.
2. Ribbon line.
3. The body.
4. The closing.

The heading usually consists of a customer name and address, date, page number and internal order or invoice number. The ribbon line can have the customer's purchase order number, order date, salesman name, terms, and delivery instructions. The body contains actual line items, including a product number, description, quantity, sales unit, price and extension. The closing area shows a subtotal, tax, add-ons, invoice total, and perhaps a promotional message.

Since invoices frequently require more than one page to list all of the line items, invoice-printing programs usually will print the complete heading on only the first page. Similarly, the closing information is usually not printed until the last page of a multiple-page invoice.

To print an invoice containing all of the above information, you would probably have a line item file, customer file, product file, and order file. The line item file should be the primary file since it must be read and indexed sequentially in order to obtain keys for the customer, product, and order files. These other files can then be referenced as secondary files.

If you assume that the line item file is already organized according to a key consisting of an internal order number and line number, a sort will not be required. Entries for physical page size, top and bottom markings, and output device will probably be required. The output device for reports that use preprinted forms should always be PRINTER. If spooled output is permitted, there will be no opportunity to load or unload the invoice forms between queued print jobs.

Forms load instructions are often best included with a forms alignment feature, which will be discussed separately. A forms unload prompt will, however, be desirable in the invoice-printing report.

Since you want to print all the line items in the item file, record selection will not be required for either primary or secondary files. Report text may be required if there are any headings not already printed on the form. Neither a report header nor a report trailer will be necessary; however, a page header must be specified.

All of the output items or the complete invoice heading and ribbon line should be included in the page header. The complete invoice closing should be specified as the page trailer. A continuation page header can be specified if you want a different heading on all invoice pages after the first page. The continuation page header might, for example, only print an order number, customer name, and page number. A continuation page trailer, if specified, will print on all pages of the invoice except the last. Since you do not want any add-ons or an invoice total to be printed until the last page, you should always specify a continuation page trailer, even if it contains only a line advance.

The Report Body structure must contain a Detail Element and a level-1 trailer. The level-1 trailer should be triggered by some field in the primary file which indicates when a new invoice is to be started. The internal order number field is a good choice for a trigger field as long as there can be no duplicate order numbers on line items from different invoices.

The Detail Element Output Items should contain all the printing and totaling specifications for line items. Any additional calculations which must be done for each line item should be specified in the Detail Element function. The level-1 trailer can be used for printing totals, add-ons, and a promotional message. These items will appear in the invoice body under the last line item. If you want these fields printed in a fixed location at the bottom of the last page, they can be included in the page trailer, as suggested above. In either case, the level-1 trailer must include a line advance of 998 as its last, or only, output item. This forces DMRUN to print the page trailer for the current invoice before starting to process line items from the next invoice.

Most documents which contain an unknown number of line items can be printed using the methods described above. In the case of some medical, tax, or personnel forms, there may not be any recognizable line items. These types of forms often can be printed by specifying a page header, Detail Element, and page trailer. This provides a maximum of 300 fields which can be printed on a single form with only one repetition of the Detail Element.

FORMS ALIGNMENT

A forms alignment feature can be provided for reports using preprinted forms. This is done by specifying a Run-time Data program which is initiated when the report is selected. The Run-time Data program should provide two menu selections. One selection should result in linkage to a special forms alignment Report program. The other menu selection should link to the actual report that was originally selected.

The Repeating functions of the Run-time Data program should be used to set X3 according to the application user's selection of either forms alignment or the report. Both programs will have to be included in the Link Information of the Run-time Data program.

The forms alignment Report program has four important specifications. First, the output device must be PRINTER. Second, only a forms load prompt need be specified. Third, the report should consist of only a Report Header Element which contains the forms alignment characters as output items. And fourth, the program should reference the Run-time Data program as the first entry in the Program/Utility Linkage section.

When the preprinted form report is selected, the Run-time Data form will appear. If the application user selects the forms alignment option, the forms load prompt of the forms alignment report will appear. When the operator indicates the forms are loaded, the alignment characters print and the Run-time Data form reappears. If the forms are not correctly aligned, the forms alignment option may be selected as many times as necessary. Once the forms are aligned, the report option is selected. When the report terminates, the unload forms prompt from the actual report will appear.

RECORD SELECTION

Record selection is implemented by using the CANCEL function statement conditionally. There are three points in the report processing cycle where record selection function statements are applied. These points are immediately after the control file is read (if the control file is not the primary file), immediately after the primary file is read, and again after all secondary files have been read.

In order to obtain the best performance of Data Manager reports, you should do record selection as soon as all the required data is available. For example, if the primary file records contain all the data used for record selection, you should use the Primary File Record Selection programming form of DMCREATE to enter your function statements. You could use the Secondary File Record Selection form for this purpose, but DMRUN would then unnecessarily read secondary file records for primary-file records which won't be selected.

If, on the other hand, you cannot do your record selection without referencing secondary-file data fields, you must use the Secondary File Record Selection form for your function statements.

CONTROL FILE

The control file is used to provide high-level record selection in report programs which reference secondary files. Normally the control file is the primary file, and your first opportunity to select records is following each read of the primary file. In some cases, however, you can produce a much more efficient report if you specify a secondary file as the control file.

Suppose you have a large transaction history file with a key of customer number and transaction date. You want to print a report of transactions for customers with year-to-date purchases of more then $10,000. The year-to-date purchase data is in a customer master file.

The transaction history file will have to be the primary file since reading it index-sequentially is the only way you can access all the transactions for a customer. The customer file will be a secondary file. The processing to produce this report will involve reading all the transactions and accessing the customer file randomly to check the year-to-date purchase data. If the year-to-date balance is less than $10,000, the output is cancelled and the next transaction record is read.

A report specified in this way will produce the desired results, but it may be very inefficient. If, for example, the first customer had 300 transaction records and a year-to-date purchase balance of $6000, DMRUN would read all 300 transaction records without finding anything to print. Since the primary file is being read index-sequentially, all the transactions must be read even though a single read of the customer file would indicate that none of the transactions would be selected.

To make this type of reporting more efficient, you should specify that the control file is the customer file. You will then also be able to specify record selection for the control file.

When your report is run, DMRUN will first access the customer file index-sequentially and apply your record selection requirements until a customer with a high enough balance is found. DMRUN will then use fields from the customer file as a partial key to access the history file. The history file will be read index-sequentially and all transactions with a matching partial key will be processed and printed. When a non-matching partial key is encountered in the history file, DMRUN will again start reading the customer file until the next customer record is selected.

By using the customer file as the control file, DMRUN reads only "good" transaction records from the history file. This concept can be applied to many reports and can make a dramatic difference in report performance.

An additional feature available when the control file is a secondary file is the printing of group headers and trailers for empty groups. In the example above, suppose you specified a level-1 header to print a customer name at the beginning of each group of transactions. As explained in the section about Grouping, if there are no transactions for a customer, there will be no header to show the customer's name. To force the printing of a group header, specify that you want headers and trailers associated with the control file. (See DMCREATE programming form 270 - Input File Record Description Programs.) DMRUN will then assume that the highest level of grouping elements specified should be printed for each selected record from the control file. This will occur whether or not there are any Detail Elements within the group. If you use this feature, be sure that you do not reference any information from any files other than the control file as output items for your header and trailer.

SORT OPTION

B20 Data Manager provides you with an optional key sequence for reports. If the existing key for a control file provides the correct sequence, you do not need to specify a sort sequence. If you want to use grouping, you may need to sort the control file such that all the records belonging in a group are accessed in succession as the file is read index-sequentially.

Note that if a range of records is specified in conjunction with a sort, the range-of-records selection is applied first (i.e., before the sort). In other words, only records in the specified range will be sorted.

See form 280 - Sort Specifications For Control File for further information.

CALCULATIONS

Calculations can be specified in any of the report functions. Calculations based on file data and resulting in intermediate report output items should not be specified in the Record Selection functions. Intermediate output items are fields printed before the report trailer. Final output items, however, can be calculated in Record Selection functions. An example would be function statements which accumulate the total number of primary-file records read during the entire report.

File data should not be referenced in the report header or trailer functions since no files have been accessed when those function statements are processed.

All other Report functions have access to the appropriate file data at all times. Header Elements have access to data in the records associated with the next Detail Element to be printed. Trailer Elements have access to data in the records associated with the last printed Detail Element.

When specifying an output item to print the result of a calculation, you can use the Output Description feature to suppress printing of insignificant digits. The contents of the Program Register containing the calculation result will be truncated to fit your Output Description.

RANGE OF RECORDS

If you do not want a Report program to process the entire control file regardless of whether it is the primary or a secondary file, you can specify a starting and/or ending key value. The report program will only access control file records within the specified range of keys.

The range of records is specified by creating a Run-time Data program with fields in which to enter the key values. The Run-time Data program should assign the value T (true) or 1 to register X19 and should provide linkage to initiate the report.

If you wish to start at the beginning of the file and process to a specific ending record, specify only an upper range. If you wish to start at a specific record and process to the end of the file, specify only a lower range. If you specify partial keys for the range of records, the partial key must be at the start of the key (i.e., a prefix).

You can also load the lower and upper keys for a range of records, independently in the Repeating function. These keys are stored in registers X17 and X18. They can be loaded using the ASSIGNMENT statement.

Note that if the range of records is to be applied to a Report program which also has a sort specified, the range-of-records selection will be applied first, (i.e., before the sort). In other words, only records in the specified range will be sorted.

Example:

```
X17 = A1
X18 = A2
```

If the key uses multiple fields, you can use the concatenation operator to construct the key.

Example:

```
X17 = A1 & A2
X17 = X17 & A3
X18 = A4 & A5
X18 = X18 & A6
```

PROCESSING CYCLE

The report processing cycle is quite complex. The following list is, at best, a rough approximation of what actually happens in a full-featured, one page report. It will, however, demonstrate some of the relationships discussed in previous sections.

1. Process Report Header Function.
2. Files Opened.
3. Sort.
4. Print Report Header.
5. Read Files and print body.
    a. Read control file.
    b. Process Control File Record Selection.
    c. Read primary file.
    d. Process Primary File Record Selection.
    e. Read all secondary files.
    f. Process Secondary File Record Selection.

g. Test for group boundaries.

h. Process Page Header Function.

i. Print Page Header.

j. Process Detail Element Function.

k. Print all group headers.

l. Print Detail Element.

m. Repeat steps a-f unless EOF in step a. Assume EOF in this example.

n. Print all group trailers.

6. Process Report Trailer Function.

7. Print Report Trailer.

8. Close all data files.

REPORT PROGRAMMING FORM SEQUENCE

The following sequence of forms appears when you create a Report program:

Form Number	Form Name
070	Program Type Selection
260	Report Program Sections
270	Input File Record Description Programs
280	Sort Specifications for Control File
290	Output Specifications
167	Control File Fields Used For Partial Primary File Key*
160	Primary File Fields Used For Secondary File Key*
450	File Record Selection
460	Primary File Record Selection
470	Secondary File Record Selection
300	Report Text
320	Report Header Output Items
480	Report Header Function
330	Report Trailer Output Items
490	Report Trailer Function
340	Page Header Output Items
500	Page Header Function
360	Continuation Page Header Output Items
520	Continuation Page Header Function
350	Page Trailer Output Items

510	Page Trailer Functions
370	Continuation Page Trailer Output Items
530	Continuation Page Trailer Function
310	Report Body Structure
380	Detail Element Output Items
540	Detail Element Function
390	Level 1 Header Output Items
400	Level 1 Trailer Output Items
410	Level 2 Header Output Items
420	Level 2 Trailer Output Items
430	Level 3 Header Output Items
440	Level 3 Trailer Output Items
200	Program/Utility Linkage
255	Application File Error Help

*Optional depending on Input File Record Description Programs.

RUN-TIME DATA PROGRAM

FILES

A Run-time Data program does not actually create or maintain a primary file. However, entries made by the application user are still referenced with the file letter A and a field number, as in other program types. A Run-time Data program can access up to eight secondary files.

FIELD ATTRIBUTE COMBINATIONS

All attribute combinations are available. The Include In Record attribute is significant even though a primary file is not created. You should enter Y for Include In Record for all fields you wish to reference in the Repeating function and/or the range-of-records specification.

ADDITIONAL FEATURES

Run-time Data programs are used to allow the entry of key values for the Range of Records feature of Batch Update and Report programs. Fields for the entry of lower and upper key values are specified in the normal manner. The lower and upper key fields are then identified as containing special range-of-records information when the appropriate programming forms appear in DMCREATE. Note that even if the key sequence of the file to which a range is applied is descending, the lower key is still specified first and the upper key second.

In the Repeating function of the Run-time Data program, you should assign the value T or 1 to register X19. Also, enter the name of the Report or Batch Update program on the Program/Utility Linkage form of DMCREATE.

You can also load the lower and upper keys for a range of records, independently in the Repeating function. These keys are stored in registers X17 and X18. They can be loaded using the ASSIGNMENT statement.

Example:

```
X17 = A1
X18 = A2
```

If the key uses multiple fields, you can use the concatenation operator to construct the key.

Example:

```
X17 = A1 & A2
X17 = X17 & A3
X18 = A4 & A5
X18 = X18 & A6
```

PROCESSING CYCLE

The following steps are executed only once in a Run-time Data program. The program terminates automatically when form processing is completed.

1. Initialize form.
2. Enter data. Build secondary keys, read secondary files as required.

   The cursor wraps around through the data fields until you press the GO key.
3. Process fields. Process all fields in program starting with selected field.
4. Process Repeating Function.
5. Build range-of-records keys (if specified).

RUN-TIME DATA PROGRAMMING FORM SEQUENCE

The following sequence of forms appears when you create a Run-time Data program:

Form Number	Form Name
070	Program Type Selection
230	Run-time Data Program Sections
100	Secondary Files Record Description Programs
110	Form Selection
120	Forms Editor Form*
130	Standard Form - Form Heading*
140	Numeric Range Table
145	Alphanumeric Range Table
150	Field Definition
169	Program Fields Used for Secondary File Key**
163	Range of Records Lower Key Fields
165	Range of Records Upper Key Fields
170	Initial Function
190	Repeating Function
200	Program/Utility Linkage
255	Application File Error Help

*Optional depending on Form Selection.
**Optional depending on Secondary Files Record Description Programs.

MENU PROGRAM

Although Menu programs are normally used only for initiating other application programs, they can also access data files. The primary file in a Menu program is optional and is an "add-only" file. It is functionally similar to the primary file of a Real-time Update program and is called a Menu Selection Log. Menu transaction information, secondary-file data, and computed items can be logged in this file. A Menu program can access up to eight secondary files. In addition, you can specify up to five indices for the file created by the menu.

FIELD ATTRIBUTE COMBINATIONS

Any attribute combinations which result in duplicate Menu Selection Log keys are invalid in a Menu program. All other combinations listed in Section 10 are valid.

ADDITIONAL FEATURES

Since Menu Selection Log records can be defined by Menu programs, these programs can be referenced as record descriptions by other Data Manager programs in your application system. A common example would be to reference a Menu program as a record description for a report. This is the easiest way to create a report which prints a journal indicating the sequence of application program operations. Application user identification could also be included in such a report if it is entered along with the menu selections.

A security system can be implemented for Data Manager application programs by creating a file containing valid operator-IDs, menu names, and menu selections. Existence checks can be done against this file to validate menu selections.

One unique feature of Menu programs is that they automatically initialize all Program Registers. This means that if you link together a series of non-menu programs, you can easily pass information between them. When the series ends and you return to the previous menu, the registers will be cleared in preparation for the next program sequence.

The contents of the last field specified in a Menu program are automatically moved to register X3 after all fields in the program have been processed. If the last field is numeric, it will automatically be used as an index to your Program/Utility Linkage specifications. If the field is not numeric, it will be ignored. You override the automatic assignment of a numeric field to X3 by making your own assignment in the Repeating function of the Menu program.

MENU SELECTION LOG FILE

Menu information to be written to a Menu program Menu Selection Log file must include a unique key. The techniques explained in the Real-time Update Transaction-Log section also apply to Menu programs.

PROCESSING CYCLE

The following steps summarize form processing in a Menu program. The program terminates automatically when form processing is completed.

1. Initalize form.

2. Get key fields.*

3. Enter other fields. Build secondary keys and read secondary files as required.

   The cursor wraps around through non-key fields until you press the GO key.

4. Process fields. Process all fields in program, starting with the selected field.

5. Process Repeating function.

6. Validate unique Menu Selection Log key.*

7. Write Menu Selection Log.

*This step occurs only if a Menu Selection Log is specified.

MENU PROGRAMMING FORM SEQUENCE

The following sequence of forms appears when you create a Menu program:

Form Number	Form Name
070	Program Type Selection
084	Menu Program Sections
120	Forms Editor Form
095	Menu Selection Log
100	Secondary Files Record Description Programs
140	Numeric Range Table
145	Alphanumeric Range Table
150	Field Definition
550	Primary File Indices**
169	Program Fields Used For Secondary File Key*
170	Initial Function
190	Repeating Function
200	Program/Utility Linkage
250	Application File Error Help

*Optional depending on Secondary Files Record Description Programs.
**Optional depending on Menu Selection Log.

STANDARD MENU

The Standard Menu program type is a special purpose feature which is intended only to provide the fastest possible way to create a Menu program. No Forms Editor form is required. DMRUN will automatically generate a form image at run time. The entire menu specification is entered on a single form (245 Standard Form Menu) in DMCREATE. Like a normal Menu program, the Standard Menu initializes Program Registers. It terminates automatically when the selection has been processed.

APPENDIX A

HOW TO UNDERSTAND DM FORMS

All programs in the B20 Data Manager system are created by entering data in forms. These forms are listed below according to the form number, followed by the name of the form. There is a short overview of each form, followed by pertinent information concerning each prompt. The areas covered will be entry, range, default values, notes of importance, and the function of each prompt.

The forms contained in this appendix are listed below:

Form Number	Form Name
010	PROGRAMMING ACTIVITY SELECTION
020	CREATE/MAINTAIN DM PROGRAM - LIBRARY AND PROGRAM INFORMATION
025	DELETE DM PROGRAM - LIBRARY AND PROGRAM INFORMATION
027	LIST DM PROGRAM - LIBRARY AND PROGRAM INFORMATION
030	DISPLAY LIBRARY DIRECTORY - LIBRARY INFORMATION
033	CREATE/MAINTAIN PROGRAM LIBRARY - LIBRARY INFORMATION
040	PROGRAM LIBRARY CONTROL INFORMATION
060	COPY DM PROGRAM
070	PROGRAM TYPE SELECTION
080	MAINTENANCE PROGRAM SECTIONS
082	REAL-TIME UPDATE PROGRAM SECTIONS
084	MENU PROGRAM SECTIONS
090	PRIMARY DATA FILE

093	FILE MANAGEMENT
095	MENU SELECTION LOG
097	TRANSACTION LOG
100	SECONDARY FILES RECORD DESCRIPTION PROGRAMS
103	SECONDARY FILES RECORD DESCRIPTION PROGRAMS
110	FORM SELECTION
120	FORMS EDITOR FORM
130	STANDARD FORM - FORM HEADING
140	NUMERIC RANGE TABLE
145	ALPHANUMERIC RANGE TABLE
150	FIELD DEFINITION
155	INQUIRY FIELD DEFINITION
160	PRIMARY FILE FIELDS USED FOR SECONDARY FILE KEY
163	RANGE OF RECORDS - LOWER KEY FIELDS
165	RANGE OF RECORDS - UPPER KEY FIELDS
167	CONTROL FILE FIELDS USED FOR PARTIAL PRIMARY FILE KEY
169	PROGRAM FIELDS USED FOR SECONDARY FILE KEY
170	INITIAL FUNCTION
180	INQUIRY FUNCTION
190	REPEATING FUNCTION
195	UPDATE FUNCTION
200	PROGRAM/UTILITY LINKAGE
210	BATCH UPDATE PROGRAM SECTIONS
220	PRIMARY FILE RECORD DESCRIPTION PROGRAM
225	PRIMARY FILE RECORD DESCRIPTION PROGRAM
230	RUN-TIME DATA PROGRAM SECTIONS

240	INQUIRY PROGRAM SECTIONS
245	STANDARD FORM MENU
250	APPLICATION FILE ERROR HELP
255	APPLICATION FILE ERROR HELP
260	REPORT PROGRAM SECTIONS
270	INPUT FILE RECORD DESCRIPTION PROGRAMS
280	SORT SPECIFICATIONS FOR CONTROL FILE
290	OUTPUT SPECIFICATIONS
300	REPORT TEXT
310	REPORT BODY STRUCTURE
320	REPORT HEADER OUTPUT ITEMS
330	REPORT TRAILER OUTPUT ITEMS
340	PAGE HEADER OUTPUT ITEMS
350	PAGE TRAILER OUTPUT ITEMS
360	CONTINUATION PAGE HEADER OUTPUT ITEMS
370	CONTINUATION PAGE TRAILER OUTPUT ITEMS
380	DETAIL ELEMENT OUTPUT ITEMS
390	LEVEL 1 HEADER OUTPUT ITEMS
400	LEVEL 1 TRAILER OUTPUT ITEMS
410	LEVEL 2 HEADER OUTPUT ITEMS
420	LEVEL 2 TRAILER OUTPUT ITEMS
430	LEVEL 3 HEADER OUTPUT ITEMS
440	LEVEL 3 TRAILER OUTPUT ITEMS
450	CONTROL FILE RECORD SELECTION
460	PRIMARY FILE RECORD SELECTION
470	SECONDARY FILE RECORD SELECTION

480	REPORT HEADER FUNCTION
490	REPORT TRAILER FUNCTION
500	PAGE HEADER FUNCTION
510	PAGE TRAILER FUNCTION
520	CONTINUATION PAGE HEADER FUNCTION
530	CONTINUATION PAGE TRAILER FUNCTION
540	DETAIL ELEMENT FUNCTION
550	PRIMARY FILE INDICES

DATA MANAGER FORMS

The following material explains the DM forms.

(010) PROGRAMMING ACTIVITY SELECTION

This screen initiates and ends all B20 DM programming. With it, you select the type of functions you want to perform.

ACTIVITY NUMBER

Entry:	1 digit.
Range:	1-7
Default:	1
Function:	This form indicates which activity you want to initiate.
Required:	Yes.

(020) CREATE/MAINTAIN DM PROGRAM

LIBRARY AND PROGRAM INFORMATION

This form appears whenever the Create/Maintain DM Program activity is selected. Entries on this form provide information that DMCREATE will use for storage and retrieval of programs.

A program size is supplied by the system at the end of the Create Program activity and at the end of the Maintain Program activity. The value displayed here indicates the approximate amount of memory DMRUN requires to run the program. File buffers and secondary-file information will require additional memory which is not included in the Program Size estimate.

PROGRAM SIZE

Entry:	None. Data is supplied by the system.
Range:	N/A.
Default:	N/A. Program Size is supplied by the system at the end of the Create Program activity and at the end of the Maintain Program activity.
Function:	This entry indicates the memory that DMRUN requires to load this program. Buffers and secondary-file information require additional memory which is not included in the Program Size estimate.
Required:	N/A.

LIBRARY FILE ID

Entry:	1-40 characters.

Range: Any existing Program Library file ID.

Default: Previously referenced Program Library.

Function: The Program Library name is the file ID of an ISAM file which is used to store Data Manager programs. Each library can contain one or more programs.

Note that a Program Library can be created by using the Create/Maintain Program Library function on the Programming Activity Selection menu.

Required: Yes.

PROGRAM NAME

Entry: 1-12 characters.

Range: Any characters, including spaces.

Default: None.

Function: The Program Name is used to store and retrieve your program from the Program Library.

NOTE

Two programs in the Program Library cannot have the same name.

Required: Yes.

(025) DELETE DM PROGRAM

LIBRARY AND PROGRAM INFORMATION

This form appears whenever the Delete DM Program activity is selected on the Programming Activity Selection menu. Entries on this form identify the DM program which will be deleted.

LIBRARY FILE ID

Entry:	1-40 characters.
Range:	Any existing Program Library file ID.
Default:	Previously referenced Program Library.
Function:	This entry identifies the Program Library containing the program to be deleted.
Required:	Yes.

PROGRAM NAME

Entry:	1-12 characters.
Range:	Any characters, including spaces.
Default:	None.
Function:	This entry identifies the DM program to be deleted.
Required:	Yes.

(027) LIST DM PROGRAM

LIBRARY AND PROGRAM INFORMATION

This form appears whenever the List DM Program activity is selected on the Programming Activity Selection menu. Entries on this form identify the DM program that will be listed.

LIBRARY FILE ID

Entry: 1-40 characters.

Range: Any existing Program Library file ID.

Default: Previously referenced Program Library.

Function: This entry identifies the Program Library containing the DM program to be listed.

Required: Yes.

PROGRAM NAME

Entry: 1-12 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry identifies the DM program to be listed.

Required: Yes.

(030) DISPLAY LIBRARY DIRECTORY

LIBRARY INFORMATION

This form appears whenever you select the Display Library Directory activity on the Programming Activity Selection menu. This form indicates the Program Library for which directory information is to be displayed. The information supplied is the DM name, type, size, and version ID. This information can be printed by pressing F8 after the directory information has been displayed.

LIBRARY FILE ID

Entry:	1-40 characters.
Range:	Any existing Program Library file ID.
Default:	Previously referenced Program Library.
Function:	This prompt indicates the Program Library for which directory information is to be displayed.
Required:	Yes.

(033) CREATE/MAINTAIN PROGRAM LIBRARY

LIBRARY INFORMATION

This form appears whenever the Create/Maintain Program Library activity is selected on the Programming Activity Selection menu. The entry on this form indicates the Program Library which is to be created or maintained.

LIBRARY FILE ID

Entry: 1-40 characters.

Range: For Maintain: Any existing Program Library file ID.

For Create: Any unique B20 file ID.

Default: Previously referenced Program Library.

Function: This entry indicates the Program Library which is to be created or maintained.

Required: Yes.

(040) PROGRAM LIBRARY CONTROL INFORMATION

This is the second form appearing in the Create/Maintain Program Library activity. These entries provide static information stored in the Program Library control record. The library can be created without making any of these entries; they can be made later with the Maintain option of this activity.

INITIAL DM PROGRAM

Entry: 1-12 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry identifies the DM program in this library which will be initiated automatically when the library is selected on the DMRUN Library Selection menu. This is normally a menu program which provides linkage to all other programs in the library.

Required: No. Programs can be accessed by the Direct Program Selection option in DMRUN.

FORMS LIBRARY FILE ID

Entry: 1-40 characters.

Range: File ID of a Forms Library.

Default: None.

Function: This entry identifies the Forms Library containing all Forms Editor entry forms and help forms referenced by DM programs in this Program Library.

Required: Not if all programs in this Program Library use only standard forms.

PROGRAM LIBRARY VERSION ID

Entry: 1-16 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry allows you to identify different versions of a Program Library. The version ID will be moved to register X21 by DMRUN whenever this library is accessed. It is then available to any program in the library and can be referenced as a field source.

Required: No.

(060) COPY DM PROGRAM

This form appears whenever the Copy DM Program activity is selected on the Programming Activity Selection menu. These entries provide DMCREATE with information about the DM program to be copied (source program) and the destination (copy to) library and program name.

SOURCE PROGRAM

LIBRARY FILE ID

Entry:	1-40 characters.
Range:	Any existing Program Library file ID.
Default:	Previously referenced Program Library.
Function:	This entry identifies the Program Library which contains the DM program to be copied.
Required:	Yes.

PROGRAM NAME

Entry:	1-12 characters.
Range:	Any existing program name.
Default:	None.
Function:	This entry identifies the DM program to be copied.
Required:	Yes.

COPY TO

LIBRARY FILE ID

Entry: 1-40 characters.

Range: Any existing Program Library file ID.

Default: Previously referenced Program Library.

Function: This entry identifies the Program Library which will contain the copied DM program.

Required: Yes.

PROGRAM NAME

Entry: 1-12 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry identifies the name used to store the copied program.

Required: Yes.

(070) PROGRAM TYPE SELECTION

This is the second form which appears whenever the Create/Maintain DM Program activity is selected on the Programming Activity Selection menu. When you are creating a new DM program, it allows you to choose the program type. You cannot change the program type once a program has been created. When using either the Create or Change option, you can enter other information on this form.

PROGRAM TYPE

Entry: 1 digit.

Range: 1-8

Default: None.

Function: This choice controls which screens appear next. Each different type of program requires different information.

Required: Yes.

PROGRAM HELP FORM

Entry: 1-4 characters.

Range: Any 4 characters, excluding spaces.

Default: None.

Function: This entry identifies a help form which should contain explanatory information about the entire program, and it can be created using the Forms Editor utility. DMRUN will display this help form when program help is requested by the application user.

NOTE

This is not used for Batch Update and Report Programs.

Required: No. If you do not make an entry here, the standard DM Program Help Form will be used.

PROGRAM ERROR HELP FORM:

Entry: 1-4 characters.

Range: Any 4 characters, excluding spaces.

Default: None.

Function: This entry indicates the name of the help form to be displayed when DMRUN encounters program errors and help is requested.

Note that program errors result from entering incorrect or inconsistent program specifications. This type of error is usually detected during the normal testing of your DM programs. In the unlikely event that an application user receives a DM program with an error in it, this help form can be used to explain the situation and suggest appropriate corrective actions.

Required: No. If an entry is not made here, the standard DM Program Error Help Form will be used.

PROGRAM VERSION ID

Entry: 1-16 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry allows you to identify different versions of a DM program. The program version ID will be moved to register X20 by DMRUN whenever this program is accessed. It is then available to the program and can be referenced as a field source.

Required: No.

(080) MAINTENANCE PROGRAM SECTIONS

This form appears when you have selected a program type on the Program Type Selection menu. It allows access to the sections of the program type you selected.

If you do not want this program in your program library, return to the Programming Activity Selection menu and select the Delete DM Program activity.

SECTION

Entry:	1 digit.
Range:	1-9
Default:	1
Function:	If you are creating a program, use the default value (1). DMCREATE will then proceed through the programming forms until all forms have been completed or until you press the FINISH key. If you are maintaining a program, enter the number of the section containing the specifications you want to change. Press FINISH when you have entered all the changes.
Required:	Yes.

(082) REAL-TIME UPDATE PROGRAM SECTIONS

This form appears when you have selected a program type on the Program Type Selection menu. It allows access to the sections of the program type you selected.

If you do not want this program in your program library, return to the Programming Activity Selection menu and select the Delete DM Program activity.

SECTION

Entry:	1 digit.
Range:	1-9
Default:	1
Function:	If you are creating a program, use the default value (1). DMCREATE will then proceed through the programming forms until all forms have been completed or until you press the FINISH key. If you are maintaining a program, enter the number of the section containing the specifications you want to change. Press FINISH when you have entered all the changes.
Required:	Yes.

(084) MENU PROGRAM SECTIONS

This form appears when you have selected a program type on the Program Type Selection menu. It allows access to the sections of the program type you selected.

If you do not want this program in your program library, return to the Programming Activity Selection menu and select the Delete DM Program activity.

SECTION

Entry:	1 digit.
Range:	1-9
Default:	1
Function:	If you are creating a program, use the default value (1). DMCREATE will then proceed through the programming forms until all forms have been completed or until you press the FINISH key. If you are maintaining a program, enter the number of the section containing the specifications you want to change. Press FINISH when you have entered all the changes.
Required:	Yes.

(090) PRIMARY DATA FILE

This is the first form of the Maintenance program Input/Output section. It allows entry of information about the file this program will create and maintain.

PRIMARY FILE ID

Entry: 1-40 characters.

Range: Any valid B20 file ID.

Default: If the volume and directory names of the file ID are not entered here, the access path will default to the path in effect when the file is accessed by DMRUN.

Function: This entry indicates the volume, directory, and file name of the data file created and maintained by this DM program.

Required: Yes.

FILE SECURITY - RECORDS ADDED

Entry: 1 character.

Range: Y or N

Default: Y

Function: This entry determines whether or not the application user will be able to add records to the primary data file with this program.

If you enter N (no) and the application user enters a nonexistent key while running this program, DMRUN will display an error message and the addition will not be allowed.

If you enter Y (yes) and the application user enters a nonexistent key while running this program, DMRUN will allow the record to be added.

Required: Yes.

FILE SECURITY - RECORDS CHANGED

Entry: 1 character.

Range: Y or N

Default: Y

Function: This entry determines whether or not this program will allow data to be changed in the primary data file. If you enter N (no) and the application user enters an existing key while this program runs, the data from the record will be displayed.

At this point, the application user can only cancel the transaction or possibly use the DELETE key (depending on the delete option below). Any attempt to change data in the record will cause DMRUN to display an error message.

Required: Yes.

FILE SECURITY - RECORDS DELETED

Entry: 1 character.

Range: Y or N

Default: Y

Function: This entry determines whether or not this program will allow records to be deleted from the primary data file. If you enter N (no) and the application user attempts to delete a record, DMRUN will display an error message and the deletion will not be allowed.

Required: Yes.

SECURITY ERROR HELP FORM

Entry: 1-4 characters.

Range: Any characters, excluding spaces.

Default: None.

Function: This entry indicates the name of the help form DMRUN displays when an application user attempts to break one of the above file security restrictions and help is requested.

Required: No. If you do not make an entry here, the standard DM Security Error Help Form will be used.

(093) FILE MANAGEMENT

This form is in the Maintenance program Input/Output section. It allows entry of a specification which determines how DMRUN will handle a file-not-found condition when this program is run.

SELECTION NUMBER

Entry: 1 digit.

Range: 1-2

Default: 2

Function: If you enter 1, DMRUN will display an error message when the primary file is not available at run time. If you use this option, you may want to provide another Maintenance program which will only create the file.

If you enter 2, DMRUN will automatically create the primary data file if it is not available at run time.

Required: Yes.

(095) MENU SELECTION LOG

This form is in the Menu program Input/Output section. You will enter information used to identify the optional menu selection log in your Menu program.

FILE ID

Entry: 1-40 characters.

Range: Any valid B20 file ID.

Default: If the volume and directory names of the file ID are not entered here, the access path will default to the path in effect when the file is accessed by DMRUN.

Function: This entry indicates the volume, directory, and file name of the log file created by this DM program.

Note that a menu selection log is an add-only file; records cannot be changed or deleted.

Required: No. If you do not enter a file name, DMRUN will assume that no log file is to be created.

FILE UNAVAILABLE

Entry: 1 digit.

Range: 1-2

Default: 2

Function: If you enter 1, DMRUN will display an error message when the primary file is not available at run time. If you use this option, you may want to provide a Maintenance program which will do nothing other than create the file.

If you enter 2, DMRUN will automatically create the primary data file if it is not available at run time.

Required: Yes.

(097) TRANSACTION LOG

This form is in the Real-time Update Program Input/Output section. An optional feature of the Real-time Update program type is the ability to create a log file containing all or selected fields from each transaction. This form allows you to name the log file.

FILE ID

Entry: 1-40 characters.

Range: Any valid B20 file ID.

Default: If you do not enter the volume and directory names of the file ID here, the access path will default to the path which is in effect when the file is accessed by DMRUN.

Function: This entry indicates the file ID of the log file which DMRUN can create to store data from each transaction.

Required: No. If you do not enter a log file name, DMRUN will assume that no log file is to be created.

(100) SECONDARY FILES RECORD DESCRIPTION PROGRAMS

This form appears in the Input/Output section of Maintenance, Inquiry, Run-time Data, and Menu programs.

If your program is to reference data in any files other than the primary file, you must enter the names of the programs which are used to create and maintain those files. You can enter the names of these programs on this form whether or not they have been created. In order to run this program, however, all programs referenced on this form will have to be available in the Program Library of this program.

PROGRAM NAME

Entry:	1-12 characters.
Range:	DM program name.
Default:	None.
Function:	This entry identifies a program that describes a file referenced by this program.
Required:	No.

ACCESS CONTROL

Entry:	1 digit.
Range:	1-2
Default:	2

Function:	If you enter 1 (read not critical) and the read of the secondary file fails, processing will continue without the record. If you enter 2 (read critical) and the read of the secondary file fails, an error message will appear on the screen.
Required:	Yes.

INDEX NUMBER

Entry:	1 digit.
Range:	1-5
Default:	1
Function:	This entry identifies which index will be used to read this file. The index selected must not allow duplicates.
Required:	Yes.

(103) SECONDARY FILES RECORD DESCRIPTION PROGRAMS

This form appears in the Input/Output section of Real-time Update and Batch Update programs. If your program is to reference fields in any files other than the primary file, you must enter the names of the programs which are used to create and maintain those files. You can enter the names of these programs on this form whether or not they have been created. In order to run this program, however, all programs referenced on this form will have to be available in the Program Library of this program.

PROGRAM NAME

Entry: 1-12 characters.

Range: DM program name.

Default: None.

Function: This entry identifies a program that describes a file referenced by this program.

Required: No.

ACCESS CONTROL

Entry: 1 digit.

Range: 1-4

Default: 2

Function: If you enter 1 (read not critical) and the read of the secondary file fails, processing will continue without the record.

If you enter 2 (read critical) and the read of the secondary file fails, an error message will appear on the screen. You cannot update any field in this file's record.

If you enter 3 (update) and the read of the secondary file fails, an error message will appear on the screen.

If you enter 4 (add) and DMRUN cannot find the record in the secondary file, the record will be added. If the record is found, an error message will appear on the screen.

Required: Yes.

INDEX NUMBER

Entry: 1 digit.

Range: 1-5

Default: 1

Function: This entry identifies which index will be used to read this file. The index selected must not allow duplicates.

NOTE

> If the access control is 4 (add), you will not be allowed to make an entry in this field.

Required: Yes.

(110) FORM SELECTION

This form is in the Input/Output section of Maintenance, Real-time Update, Inquiry, and Run-time Data programs.

All DM programs that use forms can interface to a form created by the Forms Editor. If your program does not require more than 20 fields and you do not need the features of a Forms Editor form, you can select the DM Standard Form option.

If you use a standard form, DMRUN will automatically format your form heading, prompts, and entry fields.

If you create your own form, you will use the Forms Editor and Librarian. Be sure to identify the Forms Library name in the Program Library control record by using the Create/Maintain Program Library utility.

SELECTION NUMBER

Entry:	1 digit.
Range:	1-2
Default:	1
Function:	This entry determines if a Standard Form or a Forms Editor Form will be used by your program.
Required:	Yes.

(120) FORMS EDITOR FORM

This form appears if you have selected a Forms Editor form on the Form Selection menu. It is in the Input/Output section of Maintenance, Real-time Update, Inquiry, Run-time Data, and Menu programs.

FORM NAME

Entry: 1-12 characters.

Range: Any characters, excluding spaces.

Default: None.

Function: This entry indicates which form in the Forms Library will be used by your program.

Required: Yes, if a Forms Editor form is to be used by this program.

(130) STANDARD FORM - FORM HEADING

This form appears if the Standard Form option was selected on the Form Selection menu. It allows you to enter a form heading or title which will appear centered at the top of the screen when your program is run. This form is in the Input/Output section of Maintenance, Real-time Update, Inquiry, and Run-time Data programs.

FORM HEADING

Entry: 1-48 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry provides a form heading or title which will appear at the top of the form when the program is run. The heading will automatically be centered.

Required: No.

(140) NUMERIC RANGE TABLE

This is the first form of the Ranges section. If you need to use the range-checking feature of DMRUN for validating numeric fields entered by the application user, enter all the numeric ranges you need for this program. You will be able to apply up to five of these ranges to each field you define in the next program section. The same range can be applied to several fields.

LOWER LIMIT

Entry: An optional sign and a 1-15 digit number with an optional decimal point.

Range: Any number.

Default: None.

Function: This entry indicates the lower limit of a numeric range.

Required: No.

UPPER LIMIT

Entry: An optional sign and 1-15 digit number with an optional decimal point.

Range: Any number which is greater than or equal to the lower range limit.

Default: None.

Function: This entry indicates the upper limit of a numeric range.

Required: Yes, if a lower limit was entered.

(145) ALPHANUMERIC RANGE TABLE

This is the second form of the Ranges section. If you need to use the range-checking feature of DMRUN to validate alphanumeric fields entered by the application user, enter all the alphabetic ranges you need for this program. You will be able to apply up to five of these ranges to each field you define in the Field Definitions section. The same range can apply to several fields.

LOWER LIMIT

Entry: 1-16 characters.

Range: Any characters, including spaces.

NOTE

If you select the alphanumeric range option, you must enter either an upper or a lower range limit.

Default: None.

Function: This entry indicates the lower limit of an alphabetic range.

Required: No.

UPPER LIMIT

Entry: 1-16 characters.

Range: Any characters, including spaces.

NOTE

If you select the alphanumeric range option, you must enter either an upper or a lower range limit.

Default: None.

Function: This entry indicates the upper limit of an alphabetic range.

Required: No.

(150) FIELD DEFINITION

This is the only form in the Field Definitions section. One field can be defined on this form. The form will repeat once for each field you want to define. The form stops repeating when you press GO on a form which has no entries, or until you press F3 to exit the Field Definitions section. Notice the current field number is always displayed as a part of the form heading.

The entries on these forms will define a data file record description as well as some of the data validation and operational features used by DMRUN. See Section 10 of this manual for more information on valid combinations of features you can specify.

NEXT FIELD

Entry:	1-3 digits.
Range:	1-100
Default:	None.
Function:	If you create a new program and need to go back to a previous field, position the cursor by using the Up Arrow key. Type the number of the field to which you want to return and press the RETURN key or the GO key. You also can use the NEXT FIELD entry to skip forward to any succeeding field that has already been defined.
Required:	No.

FIELD NAME

Entry:	1-16 characters.

Range: If a Forms Editor form is being used, the field name should not contain any spaces other than trailing spaces.

Default: None.

Function: If your program uses a Forms Editor form and this field is to be included on the form, this entry should be the name of a field on that form. If your program uses a Standard Form, your field name will appear on the Standard Form as a prompt.

Required: No. If the FIELD NAME field is left blank, an entry or display of data will not be permitted for this field.

STD DESCRIPTION

Entry: 1-2 digits.

Range: 1-28

Default: None.

Function: This number indicates the field type, length, and decimal placement. The chart on the right of the Field Definition form shows the meaning of the 28 standard field descriptions. Notice that Description 14 is for dates only. It will trigger automatic date validation in DMRUN.

If you select Standard Descriptions 26-28, you are also required to enter a DESCRIPTION to tell DMRUN the display format of this field.

Required: No. You can use either the STD DESCRIPTION or the DESCRIPTION entry to describe any field except a date field.

DESCRIPTION

Entry: 1-5 characters.

Range: Numeric fields have a maximum length of 15 digits. The maximum alpha field is 45 characters.

Use the following symbols when specifying a field:

+, -, or S for sign.

Digits 0-9.

A period (.) for decimal point.

Examples:

Entry	Field Description
6	A 6-character alphanumeric field.
8.0	An 8-digit unsigned whole number.
+8.0	An 8-digit signed whole number.
4.2	A 6-digit unsigned number with two decimal places.

Default: None.

Function: This entry can be used to specify field descriptions which are not available as standard descriptions. A field description indicates type (alpha or numeric) by the presence of a decimal point in all numeric descriptions. The sign attribute is specified by adding a plus sign (+), a minus sign (-), or the letter S as a prefix to numeric descriptions.

Whole number places precede the decimal point. The number of decimal places is entered after the decimal point.

Required: Yes, if Standard Descriptions 26-28 were specified.

MANDATORY ENTRY

Entry: 1 character.

Range: Y or N

Default: N

Function: If you enter a Y (yes) when describing an alpha field, the application user will not be allowed to make a blank entry. If the field being described is numeric, neither blank nor zero entries will be allowed.

Required: Yes.

INCLUDE IN RECORD

Entry: 1 character.

Range: Y or N

Default: Y

Function: This entry determines whether or not this field will be included in the record description for the primary file. Most fields will be included in the primary file record description unless, for example, the field appears on the screen only for visual verification by the application user.

Any fields not included in the record will not be available either as a source or for calculations later in your program. Also, such fields cannot be automatically duplicated using the FIELD DUP/INCREMENT feature.

Required: Yes.

INCLUDE ON FORM

Entry: 1 character.

Range: Y or N

Default: Y

Function: This entry determines whether or not DMRUN will attempt to read anything in or write anything to this field on the form. If the field is not included on the form, it can still be included in the record.

Note that on a standard form a blank field name will not generate a blank line unless you have entered Y (yes) for the Include On Form feature.

Required: Yes.

ALLOW ENTRY

Entry: 1 character.

Range: Y or N

Default: Y

Function: If this field is included on the form, you can also indicate whether the application user will be allowed to make an entry in this field. If this field is being used to display information which the application user is not permitted (or does not need) to change, enter N (no).

Required: Yes.

HELP FORM NAME

Entry: 1-4 characters.

Range: Any 4 characters, excluding spaces.

Default: None.

Function: This entry identifies the optional Forms Editor form which you can create to provide information about this field in terms which are meaningful to the application user. This form will be displayed by DMRUN if the HELP key is pressed while the cursor is in this field.

Required: No. If you do not make an entry here, the standard DM Field Help Form will be used.

SOURCE

Entry: 1 character and 3 digits.

Range: Any primary- or secondary-file field identifier defined in the program or any register identifier.

Examples:

Entry	Description
B11	Secondary-file field identifier.
A2	Primary-file field identifier for a preceding field.
P6	Program register.
X1	System date X Register.

Default: None.

Function: This entry allows you to specify that a field should be initialized to show data from a secondary file, the primary file (data taken from a preceding field), a Program Register, or an X

Register. The initialization will take place when the cursor enters the field.

When information is to be taken from a secondary file, all key fields for that file must have been entered as a result of processing previous fields in your application program.

Required: No.

SKIP TO FIELD

Entry: 1-3 digits.

Range: 1-100. This entry must be greater than the number of the current field.

Default: None.

Function: This entry indicates the number of the field which DMRUN will process next if the application user makes a blank entry for this field. The skip will occur only when a record is being added and only the first time this field is entered in each transaction.

Required: No.

FIELD DUP/INCREMENT

Entry: 1 character.

Range: Y or N

Default:	N
Function:	This entry indicates whether the application user's entry in this field should be used as a default for this field in succeeding transactions when adding records to this file. This entry is not used when existing records are being modified.
Required:	Yes.

INCREMENT AMOUNT

Entry:	1-4 digits and a sign.
Range:	-999 to +9999
Default:	0
Function:	When field duplication has been specified for a numeric field, you have the additional option of specifying an increment amount. The last entry in this field plus the increment becomes the default value for the current transaction.
Required:	No.

RANGE CHECKING

Entry:	1-2 digit number for each of the five ranges.
Range:	The number of a range-table entry for the appropriate data type.

Default: None.

Function: The numbers entered in the range fields tell DMRUN which range-table entries should be used to validate information in the field being defined. If the information does not fall within at least one of the ranges specified, a range-check error is displayed to the application user and the information must be reentered.

Required: No.

(155) INQUIRY FIELD DEFINITION

This is the only form in the Field Definitions section for Inquiry programs. One field can be defined on this form. The form repeats once for each field on which you want to inquire. The form stops repeating when you press GO on a form which has no entries, or until you press F3 to exit the Field Definitions section. Notice that the current field number is always displayed as a part of the form heading.

The most important entries on this form are the FIELD NAME, SOURCE, STD DESCRIPTION (or DESCRIPTION), and INQUIRY KEY. A source is always required, and in most cases, specification of a field name is also necessary. An entry for one of the description fields is required only if the source is a numeric program register identifier. The inquiry key attribute tells DMRUN that a field must be entered before the primary file can be accessed.

The remaining entries on the FIELD DEFINITION form apply only to inquiry key fields because these are the only fields the application user enters. All other fields are returned from files accessed by the inquiry. In some inquiries, you may want to apply range checking or other validation to inquiry key fields. You can specify those requirements on this form.

NEXT FIELD

Entry:	1-3 digits.
Range:	1-100
Default:	None.
Function:	If you create a new program and need to go back to a previous field, position the cursor by using the Up Arrow key. Type the number of the field to which you want to return and press the RETURN key or the GO key. You can also use the NEXT FIELD entry to skip forward to any succeeding field that has already been defined.

Required: No.

FIELD NAME

Entry: 1-16 characters.

Range: If a Forms Editor form is being used, the field name should not contain any spaces other than trailing spaces.

Default: None.

Function: If your program uses a Forms Editor form and this field is to be included on the form, this entry should be the name of a field on that form. If your program uses a Standard Form, your field name will appear on the Standard Form as a prompt.

Required: No. If this entry is left blank, no entry or display of data will be permitted for this field.

SOURCE

Entry: 1 character and 3 digits.

Range: Any primary- or secondary-file field identifier defined in the program or any register identifier.

Examples:

Entry	Description
B11	Secondary-file field identifier.
A2	Primary-file field identifier for a preceding field.
P6	Program register.
X1	System date X Register.

Default: None.

Function: This entry allows you to specify that a field should be initialized to show data from a secondary file, the primary file, a Program Register, or an X Register. The initialization takes place after the key fields have been entered by the application user.

Required: Yes.

STD DESCRIPTION

Entry: 2 digits.

Range: 12-25

Default: None.

Function: This number indicates the field type, length, and decimal placement. The chart on the right of the Field Definition form shows the meaning of the 25 standard field descriptions. Notice that Description 14 is for dates only. It will trigger automatic date validation in DMRUN.

Required: No. You can use either the STD DESCRIPTION or the DESCRIPTION entry to describe a field. A description of either type is required only when the source of this field is a numeric program register identifier.

DESCRIPTION

Entry: 1-5 characters.

Range: Numeric fields have a maximum length of 15 digits. The maximum alpha field is 45 characters.

Use the following symbols when specifying a field:

+, -, or S for sign.

Digits 0-9 for places.

A period (.) for decimal point.

Examples:

Entry	Field Description
6	A 6-character alphanumeric field.
8.0	An 8-digit unsigned whole number.
+8.0	An 8-digit signed whole number.
4.2	A 6-digit unsigned number with two decimal places.

Default: None.

Function: This entry can be used to specify field descriptions which are not available as standard descriptions. A field description indicates type (alpha or numeric) by the presence of a decimal point in all numeric descriptions. The sign attribute is specified by adding a plus sign (+), a minus sign (-), or the letter S as a prefix to numeric descriptions.

Whole number places precede the decimal point. The number of decimal places is entered after the decimal point.

Required: No. You can use either the STD DESCRIPTION or DESCRIPTION entry to describe a field. A description of either type is required only when the source for this field is a numeric program register identifier.

MANDATORY ENTRY

Entry: 1 character.

Range: Y or N

Default: N

Function: If you enter a Y (yes) when describing an alpha field, the application user will not be allowed to make a blank entry. If the field being described is numeric, neither blank nor zero entries will be allowed.

Required: Yes.

INQUIRY KEY

Entry 1 character.

Range: Y or N

Default: N

Function: If this field is a part of the key to the primary file, enter Y (yes). At run time, all primary key fields will be processed before DMRUN attempts to read the primary file.

Required: Yes.

INCLUDE IN RECORD

Entry: 1 character.

Range: Y or N

Default: Y

Function: This entry determines whether or not this field will be included in the record description for the primary file. Most fields will be included in the primary file record description unless, for example, the field appears on the screen only for visual verification by the application user. Any fields not included in the record will not be available either as a source or for calculations later in your program.

Required: Yes.

INCLUDE ON FORM

Entry: 1 character.

Range: Y or N

Default: Y

Function: This entry determines whether or not DMRUN will attempt to read anything in or write anything to this field on the form. If the field is not included on the form, it can still be included in the record.

Note that on a standard form a blank field name will not generate a blank line unless you enter Y (yes) for the Include On Form feature.

Required: Yes.

ALLOW ENTRY

Entry: 1 character.

Range: Y or N

Default: Y

Function: If this field is included on the form, you can also indicate whether or not the application user will be allowed to make an entry in this field. If this field is being used to display information which the application user is not permitted (or does not need) to change, enter N (no).

Required: Yes.

HELP FORM NAME

Entry: 1-4 characters.

Range: Any 4 characters, excluding spaces.

Default: None.

Function: This entry identifies the optional Forms Editor form which you can create to provide information about this field in terms which are meaningful to the application user. This form will be displayed by DMRUN if the HELP key is pressed while the cursor is in this field.

Required: No. If you do not make an entry here, the standard DM Field Help Form will be used.

SKIP TO FIELD

Entry: 1-3 digits.

Range: 1-100. This entry must be greater than the number of the current field.

Default: None.

Function: This entry indicates the number of the field which DMRUN will process next if the application user makes a blank entry for this field.

Required: No.

FIELD DUP/INCREMENT

Entry: 1 character.

Range: Y or N

Default: N

Function: This entry indicates whether the application user's entry in this field should be used as a default for this field in succeeding transactions.

Required: Yes.

INCREMENT AMOUNT

Entry: 1-4 digits and a sign.

Range: -999 to +9999

Default: 0

Function: When field duplication has been specified for a numeric field, you have the additional option of specifying an increment amount. The increment amount is the default for the first transaction. The last entry plus the increment then becomes the default for this field in succeeding transactions.

Required: No.

RANGE CHECKING

Entry: 1-2 digit number for each of the five ranges.

Range: The number of a range-table entry for the appropriate data type.

Default: None.

Function: The numbers entered in the range fields tell DMRUN which range-table entries should be used to validate information in the field being defined. If the information does not fall within at least one of the ranges specified, a range-check error is displayed to the application user and the information must be reentered.

Required: No.

(160) PRIMARY FILE FIELDS USED FOR SECONDARY FILE KEY

This form appears only if secondary-file program names were specified in the Input/Output section of Batch Update, Report, and Inquiry programs. It repeats once for each secondary file you specified. Up to 16 fields in the primary-file record can be designated as parts of the key for each secondary file. You can use the POSITION IN RECORD and LENGTH OF DATA entries on this form to identify key information which cannot be defined in terms of whole fields.

FIELD NUMBER

Entry:	1-3 digits.
Range:	Must be the number of a field in the primary file record (not the number of a field in this program).
Default:	None.
Function:	This entry identifies a field that should be used as part of the key to a secondary file.
Required:	Yes, unless the key is specified in terms of POSITION IN RECORD and LENGTH OF DATA entries.

POSITION IN RECORD

Entry:	1-4 digits.
Range:	Any number less than or equal to the record size (in bytes) of the record defined by your program. The first byte is considered position 1.

Default: None.

Function: This entry indicates the beginning byte position of the secondary-file key information in the primary-file record.

Required: No, except when the key cannot be specified by field number.

LENGTH OF DATA

Entry: 1-2 digits.

Range: Any length which does not go beyond the end of the field which contains the position specified above.

Default: None.

Function: The length (in bytes) of the secondary-file key information in the primary-file record.

Required: No, except when the key cannot be specified by field number.

(163) RANGE OF RECORDS - LOWER KEY FIELDS

This form appears in the Range of Records section of a Run-time Data program. It allows you to select up to 16 of the fields specified in the Run-time Data program. The contents of these fields are used by DMRUN to construct a key value for a succeeding Report or Batch Update program. The key value is used to identify the lowest key of a range. This key will identify either the first record to be read if the index is ascending or the last record to be read if the index is descending.

You do not need to make any entries on this form unless you want to use the range of records feature. If you want to specify a range of records, be sure that register X19 contains the value "1" or "T". Register X19 can be set with a function statement in either function of this program or an Initial Function statement in the succeeding program.

You can use the POSITION IN RECORD and LENGTH OF DATA entries on this form to identify key information which cannot be defined in terms of whole fields.

FIELD NUMBER

Entry: 1-3 digits.

Range: Must be a field number defined in the Field Definition section.

Default: None.

Function: This entry indicates that a field should be used as part of a key to indicate the starting record of a range of records.

Required: Yes, if the range of records feature is used.

POSITION IN RECORD

Entry: 1-4 digits.

Range: Any number less than or equal to the sum of the lengths stored by this Run-time Data program. A field must have the Include in Record attribute to be included in this total. The first byte is considered position 1.

Default: None.

Function: This entry indicates the beginning byte position of the key field in the information stored by this Run-time Data program.

Required: No, except when the key cannot be specified by field number.

LENGTH OF DATA

Entry: 1-2 digits.

Range: 1-45

Default: None.

Function: The length (in bytes) of the key-field information stored by this Run-time Data program.

Required: No, except when the key cannot be specified by field number.

(165) RANGE OF RECORDS - UPPER KEY FIELDS

This form appears in the Range of Records section of a Run-time Data program. It allows you to select up to 16 of the fields specified in the Run-time Data program. The contents of these fields are used by DMRUN to construct a key value for a succeeding Report or Batch Update program. The key value is used to identify the highest key of a range. This key will identify either the last record to be read if the index is ascending or the first record to be read if the index is descending.

You do not need to make any entries on this form unless you want to use the range of records feature. If you want to specify a range of records, be sure that register X19 contains the value "1" or "T". Register X19 can be set with a function statement in either function of this program or an Initial Function statement in the succeeding program.

You can use the POSITION IN RECORD and LENGTH OF DATA entries on this form to identify key information which cannot be defined in terms of whole fields.

FIELD NUMBER

Entry:	1-3 digits.
Range:	Must be a field number defined in the Field Definition section.
Default:	None.
Function:	This entry indicates that a field should be used as part of a key to indicate the ending record of a range of records.
Required:	Yes, if the range of records feature is used.

POSITION IN RECORD

Entry: 1-4 digits.

Range: Any number less than or equal to the sum of the lengths stored by this Run-time Data program. A field must have the Include in Record attribute to be included in this total. The first byte is considered position 1.

Default: None.

Function: This entry indicates the beginning byte position of the key field in the information stored by a Run-time Data program.

Required: No, except when the key cannot be specified by field number.

LENGTH OF DATA

Entry: 1-2 digits.

Range: 1-45

Default: None.

Function: The length (in bytes) of the key-field information stored by a Run-time Data program.

Required: No, except when the key cannot be specified by field number.

(167) CONTROL FILE FIELDS USED FOR PARTIAL PRIMARY FILE KEY

This form appears in the Key Fields section of a Report program if the control file is not A. You can specify up to 16 fields in the control file to make the key or partial key into a primary file. If the fields specified form a partial key, these fields must be the leading part of the key in the primary file.

You can use the POSITION IN RECORD and LENGTH OF DATA entries on this form to identify key information which cannot be defined in terms of whole fields.

FIELD NUMBER

Entry: 1-3 digits.

Range: Must be the number of a field in the primary-file record.

Default: None.

Function: This entry indicates that this field should be used as part of the key to a secondary file.

Required: Yes, unless the key is specified in terms of position and length.

POSITION IN RECORD

Entry: 1-4 digits.

Range: Any number less than or equal to the record size (in bytes) of the control file. The first byte is considered position 1.

Default: None.

Function: This entry indicates the beginning byte position of the secondary-file key information in the primary file record.

Required: No, except when the key cannot be specified by field number.

LENGTH OF DATA

Entry: 1-2 digits.

Range: 1-45

Default: None.

Function: The length (in bytes) of the secondary-file key information in the primary-file record.

Required: No, except when the key cannot be specified by field number.

(169) PROGRAM FIELDS USED FOR SECONDARY FILE KEY

This form appears only if secondary-file program names were specified in the Input/Output section of Maintenance, Real-time Update, Menu, or Run-time Data programs. It repeats once for each secondary file you specified. Up to 16 fields in this program can be designated as parts of the key for each secondary file. You can use the POSITION IN RECORD and LENGTH OF DATA entries on this form to identify key information which cannot be defined in terms of whole fields.

FIELD NUMBER

Entry:	1-3 digits.
Range:	Must be the field number of a field defined in the Field Definitions section.
Default:	None.
Function:	This entry identifies a field that should be used as part of the key to a secondary file.
Required:	Yes, unless the key is specified in terms of POSITION IN RECORD and LENGTH OF DATA entries.

POSITION IN RECORD

Entry:	1-4 digits.
Range:	Any number less than or equal to the record size (in bytes) of the record defined by your program. The first byte is considered position 1.

Default: None.

Function: This entry indicates the beginning byte position of the secondary-file key information in the primary-file record.

Required: No, except when the key cannot be specified by field number.

LENGTH OF DATA

Entry: 1-2 digits.

Range: Any length which does not go beyond the end of the field which contains the position specified above.

Default: None.

Function: The length (in bytes) of the secondary-file key information in the primary-file record.

Required: No, except when the key cannot be specified by field number.

(170) INITIAL FUNCTION

This form provides access to the special function features of DMRUN. Entries on this form are not required. Any entries you make on this form should follow the guidelines presented in the Function Statements section of this manual. These function statements will be performed only once at the start of your DM program. You have access to all system and program registers but no files have been read. The Initial function can contain up to 120 function statements.

(180) INQUIRY FUNCTION

This form provides access to the special function features of DMRUN. Entries on this form are not required. Any entries you make on this form should follow the guidelines presented in the Function Statements section of this manual. These function statements will be performed for each transaction after the primary key fields have been entered and all files have been read, but before any fields are displayed. You have access to all system and program registers plus all primary and any secondary field identifiers. The Inquiry function can contain up to 120 function statements.

(190) REPEATING FUNCTION

This form provides access to the special function features of DMRUN. Entries on this form are not required. Any entries you make on this form should follow the guidelines presented in the Function Statements section of this manual. These function statements will be performed for each transaction after all files have been read, but before any files are rewritten. You have access to all system and program registers plus all primary- and secondary-file records read during the transaction. The Repeating function can contain up to 120 function statements.

(195) UPDATE FUNCTION

This form provides access to the special function and update features of DMRUN. Any entries you make on this form should follow the guidelines in the Function Statements section of this manual. These function statements will be performed for each transaction in the primary file. They are performed after all files have been read, but before any files are rewritten. You have access to all system and program registers plus all primary- and secondary-file records read during the transaction. The Update function can contain up to 120 function statements.

(200) PROGRAM/UTILITY LINKAGE

Your entries on this form can be used to determine the program that will be initiated automatically when the current DM program ends.

PROG/FILE NAME

Entry: 1-12 characters.

Range: The range can consist of the following:

1. Any DM program in the same Program Library in which you are currently working.
2. Any Assembler, BASIC, COBOL, FORTRAN, or PASCAL program.
3. The name of a record description program for the ISAM Reorganize utility.
4. The file ID of a Submit file for the Submit utility.

Default: None.

Function: This entry allows program names to be listed for automatic initiation at the end of the current program.

Required: No, unless your program is a menu.

TYPE

Entry: 1 digit.

Range: 1-8

Default: 1

Function: This entry indicates whether the PROG/FILE NAME entry is

1. An Assembler, BASIC, COBOL, DM, FORTRAN, or PASCAL program.
2. The name of a record description program for the ISAM Reorganize utility.
3. The name of a file for the Submit utility.

Required: Yes, whenever a program name is entered in the preceding field.

Note that if your program is a menu, DMRUN will use the value entered in the last field defined on the menu as an index into the list of program names on this form. The program name which corresponds to that value will be initiated. This value can be overridden by setting register X3 in the Repeating function.

If your program is not a menu, linkage to the next program can be determined as follows:

1. If no entries are made on this form, the next program will always be the last menu program that preceded the current program.
2. If one program name is specified on this form, that program will always be initiated next.
3. If several program names are listed on this form, DMRUN initiates the next program by

using register X3 as an index into your program list. You can set register X3 dynamically by making an assignment with a function statement in your program. If you do not set register X3, then the next program will be the first program listed on this screen.

(210) BATCH UPDATE PROGRAM SECTIONS

This form appears when you have selected a program type on the Program Type Selection menu. It allows access to the sections of the program type you selected.

If you do not want this program in your program library, return to the Programming Activity Selection menu and select the Delete DM Program activity.

SECTION

Entry:	1 digit.
Range:	1-6
Default:	1
Function:	If you are creating a program, use the default value (1). DMCREATE will then proceed through all of the programming forms until all forms have been completed or until you press the FINISH key. If you are maintaining a program, enter the number of the section containing the specifications you want to change. Press FINISH when you have entered all the changes.
Required:	Yes.

(220) PRIMARY FILE RECORD DESCRIPTION PROGRAM

This form identifies the DM program which defines the primary data file for an Inquiry program.

PROGRAM NAME

Entry:	1-12 characters.
Range:	The name of any Maintenance, Menu, or Real-time Update program that is in the same Program Library as the program being created. Note that if a Menu program is specified, it must have a menu selection log associated with it. If a Real-time Update program is specified, it must have a transaction log.
Default:	None.
Function:	This entry identifies the program which defines the primary data file for an Inquiry program.
Required:	Yes.

INDEX NUMBER

Entry:	1 digit.
Range:	1-5
Default:	1

Function: This entry identifies which index will be used to read the primary data file.

Required: Yes.

(225) PRIMARY FILE RECORD DESCRIPTION PROGRAM

This form identifies the DM program which defines the primary data file for a Batch Update program.

PROGRAM NAME

Entry: 1-12 characters.

Range: The name of any Maintenance, Menu, or Real-time Update program that is in the same Program Library as the program being created.

Default: None.

Function: This entry identifies a DM program which defines the primary data file.

Required: Yes.

INDEX NUMBER

Entry: 1 digit.

Range: 1-5

Default: 1

Function: This entry identifies which index will be used to read the primary data file.

Required: Yes.

DELETE PRIMARY FILE

Entry: 1 character.

Range: Y or N

Default: N

Function: This entry indicates whether or not DMRUN should automatically delete the primary file after all transactions have been processed.

Required: Yes.

(230) RUN-TIME DATA PROGRAM SECTIONS

This form appears when you have selected a program type on the Program Type Selection menu. It allows access to the sections of the program type you selected.

If you do not want this program in your Program Library, return to the Programming Activity Selection menu and select the Delete DM Program activity.

SECTION

Entry: 1 digit.

Range: 1-9

Default: 1

Function: If you are creating a program, use the default value (1). DMCREATE will then proceed through all of the programming forms until all forms have been completed or until you press the FINISH key.

If you are maintaining a program, enter the number of the section containing the specifications you want to change. Press FINISH when you have entered all the changes.

Required: Yes.

(240) INQUIRY PROGRAM SECTIONS

This form appears when you have selected a program type on the Program Type Selection menu. It allows access to the sections of the program type you selected.

If you do not want this program in your Program Library, return to the Programming Activity Selection menu and select the Delete DM Program activity.

SECTION

Entry:	1 digit.
Range:	1-8
Default:	1
Function:	If you are creating a program, use the default value (1). DMCREATE will then proceed through all of the programming forms until all forms have been completed or until you press the FINISH key. If you are maintaining a program, enter the number of the section containing the specifications you want to change. Press FINISH when you have entered all the changes.
Required:	Yes.

(245) STANDARD FORM MENU

This is the only form that is used to create a Standard Form Menu program. There is room for 15 PROG/FILE NAME and DESCRIPTION entries.

If you need to initiate more than 15 programs or to enter information on the menu form other than the menu selection itself, create a Menu rather than a Standard Form Menu.

If you do not want to complete this Standard Form Menu program, return to the Programming Activity Selection screen and use the Delete DM Program activity to delete this program.

HEADING

Entry: 1-48 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry provides a form heading or title which will appear at the top of the menu form when the program is run. The heading will automatically be centered.

Required: No.

SELECTION PROMPT

Entry: 1-16 characters.

Range: Any characters, including spaces.

Default: SELECTION NUMBER

Function: This entry identifies the field on the menu in which the application user will enter a selection number.

Required: No.

ENTRY HELP FORM

Entry: 1-4 characters.

Range: Any characters, excluding spaces.

Default: None.

Function: If you would like to create a Forms Editor form containing help information about the selections on your menu, enter the name of the form here. This help form will be displayed if the application user presses the HELP key while running this program.

Required: No. If you do not make an entry here, the standard DM Field Help Form will be displayed when the application user requests help.

PROG/FILE NAME

Entry: 1-12 characters.

Range: The range can be any of the following:

1. Any DM program in the same Program Library in which you are currently working.
2. Any Assembler, BASIC, COBOL, FORTRAN, or PASCAL program.
3. The name of a record description program for the ISAM Reorganize utility.
4. The file ID of a Submit file for the Submit utility.

Default: None.

Function: The PROG/FILE NAME entry identifies the various programs that the application user can initiate from this menu. All DM programs to which this menu refers must be in the same Program Library as this menu.

The numbers beside the PROG/FILE NAME field correspond to the selection numbers which appear on the menu form when this program is run. If the application user enters 1 for the selection number, the first program you enter here will be initiated.

Required: Yes, for each description which is to appear on the menu.

DESCRIPTION

Entry: 1-48 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry appears on the Standard Form Menu and identifies the corresponding program entered under PROG/FILE NAME.

Required: No.

TYPE

Entry: 1 digit.

Range: 1-8

Default: 1

Function: The Standard Form Menu can initiate

1. An Assembler, BASIC, COBOL, DM, FORTRAN, or PASCAL program.
2. The name of a record description program for the ISAM Reorganize utility.
3. The name of a file for the Submit utility.

This entry determines which kind of program or utility DMRUN will initiate.

Required: Yes.

(250) APPLICATION FILE ERROR HELP

This form is in the Error Help Form Names section of Batch Update, Maintenance, Menu, and Real-time Update programs. It will repeat once for each application file accessed by the application program. The file identifier (A-I) appears in the heading.

If DMRUN detects one of the file-related errors listed on this form, it will:

1. Terminate the DM application program.
2. Initiate the last previous menu program.
3. Display an error message.

If the application user presses the HELP key while the error message is displayed, the help form specified here will appear.

HELP FORM NAME

Entry: 1-4 characters.

Range: Any characters, excluding spaces.

Default: None.

Function: This entry identifies the form name containing file error help information for the file identified in the heading of this form.

Required: No. If you do not make an entry here, the standard DM Error Help Form will be used.

(255) APPLICATION FILE ERROR HELP

This form is in the Error Help Form Names section of Inquiry, Run-time Data, and Report programs. It will repeat once for each application file accessed by the application program. The file identifier (A-I) appears in the heading.

If DMRUN detects one of the file-related errors listed on this form, it will:

1. Terminate the DM application program.
2. Initiate the last previous menu program.
3. Display an error message.

If the application user presses the HELP key while the error message is displayed, the help form specified here will appear.

HELP FORM NAME

Entry: 1-4 characters.

Range: Any characters, excluding spaces.

Default: None.

Function: This entry identifies the form name containing file error help information for the file identified in the heading of this form.

Required: No. If you do not make an entry here, the standard DM Error Help Form will be used.

(260) REPORT PROGRAM SECTIONS

This form appears when you have selected a program type on the Program Type Selection menu. It allows access to the sections of the program type you selected.

If you do not want this program in your Program Library, return to the Programming Activity Selection menu and select the Delete DM Program activity.

SECTION

Entry: 1 digit.

Range: 1-9

Default: 1

Function: If you are creating a program, use the default value (1). DMCREATE will then proceed through the programming forms until all forms have been completed or until you press the FINISH key.

If you are maintaining a program, enter the number of the section containing the specifications you want to change. Press FINISH when you have entered all the changes.

Required: Yes.

(270) INPUT FILE RECORD DESCRIPTION PROGRAMS

This form is in the Input/Output section of Report programs. It is used to identify the primary file, the control file, and any secondary files which will be accessed by the report.

The program corresponding to file A defines the primary file for this report. Files B thru I are optional secondary files.

PROGRAM NAME

Entry: 1-12 characters.

Range: The name of any Maintenance, Menu, or Real-time Update program that is in the same Program Library as the program being created.

Note that if a Menu program is specified, it must have a menu selection log. If a Real-time Update program is specified, it must have a transaction log.

Default: None.

Function: This entry identifies the program which defines the data file that you will use for the report.

Required: No.

ACCESS CONTROL

Entry: 1 digit.

Range: 1-2

Default: 2

Function: If you enter 1 (read not critical) and the read of the secondary file fails, processing will continue without the record.

If you enter 2 (read critical) and the read of the secondary file fails, the report will go to end of job.

Required: Yes.

INDEX NUMBER

Entry: 1 digit.

Range: 1-5

Default: 1

Function: This entry identifies which index will be used to read this file. The index selected must not allow duplicates, unless the file is the control file or primary file.

Required: Yes.

CONTROL FILE FOR THIS REPORT

Entry: 1 character.

Range: Any file identifier letter for which a Maintenance program name was previously entered.

Default: A

Function: When the Data Manager report is being processed, DMRUN reads the control file and then selects records. Normally, the information used to select primary-file records will be in the primary-file record. In this case, enter A to make the primary file the control file.

In some reports record selection is more efficient if secondary-file information is used. In this case, identify the secondary file (B-I) as the control file.

When the control file is a secondary file, DMRUN first reads the secondary file until a record is selected. DMRUN then processes primary-file records related to the secondary-file record. When all related primary-file records have been processed, DMRUN continues reading the secondary file until another record is selected.

Required: Yes.

HEADER OR TRAILER BASED ON THE CONTROL FILE

Entry: 1 character.

Range: Y or N

Default: N

Function: If the control file is not the primary file, you can print group headers and/or trailers containing control-file information, even if there are no detail elements in the group. If you enter Y (yes), the highest level header and/or trailer will be associated with the control file and will always be printed.

If you do not want a header and/or trailer associated with the control file, enter N (no). Headers and trailers will then be printed only when there are detail elements in the group.

Required: Yes.

(280) SORT SPECIFICATIONS FOR CONTROL FILE

This form is in the Input/Output section of Report programs.

If the existing key for the control file provides the correct sequence for the report, no entries need to be made on this form.

Entries on this form indicate that index-sequential access of the control file will not produce the desired report sequence. In other words, an additional key sequence must be provided for this report. A maximum of 16 fields containing a maximum total of 64 characters of data can be specified for the report key.

FIELD NUMBER

Entry: 1-3 digits.

Range: 1-100

Default: None.

Function: This prompt indicates a field which will be included in the report key.

Required: No. To specify key data, enter selections on the POSITION IN RECORD and LENGTH OF DATA entries on this form.

POSITION IN RECORD

Entry: 1-4 digits.

Range: 1-1017 bytes

Default: None.

Function: This entry defines the field that will be included in the report key. This entry has the same function as the preceding FIELD NUMBER entry, but can be used when the data location cannot be specified in terms of existing field numbers.

Required: Yes, if a sort is required and a FIELD NUMBER is not specified.

LENGTH OF DATA

Entry: 1-2 digits.

Range: 1-45 bytes

Default: None.

Function: This entry defines the field that will be included in the report key. This entry has the same function as the preceding FIELD NUMBER entry, but can be used when the data location cannot be specified in terms of existing field numbers.

Required: Yes, if a sort is required and a FIELD NUMBER is not specified.

ASCENDING?

Entry: 1 character.

Range: Y or N

Default: Y

Function: If you enter Y (yes), this field will be sorted in ascending order.

If you enter N (no), this field will be sorted in descending order.

Required: Yes.

(290) OUTPUT SPECIFICATIONS

This form is in the Input/Output section of Report programs. Entries on this form provide miscellaneous information to DMRUN.

PHYSICAL PAGE SIZE

Entry:	1-3 digits.
Range:	1-255
Default:	66
Function:	This entry indicates the page length. If the output device is to be the screen, this entry should be one less than the screen size. For example, enter 33 on a B22 with a screen size of 34.
Required:	Yes.

BLANK LINES AT TOP OF PAGE

Entry:	1-3 digits.
Range:	0-255
Default:	6
Function:	This entry indicates the size of the top margin of a report page.
Required:	No.

BLANK LINES AT BOTTOM OF PAGE

Entry: 1-3 digits.

Range: 0-255

Default: 6

Function: This entry indicates the number of blank lines remaining at the bottom of a report page.

Required: No.

OUTPUT DEVICE

Entry: 1 digit.

Range: 1-5

Default: 4

Function: The output device for the report can be specified here or determined at run time by a report header function statement in the Report program or by a function statement in a preceding program. The function statement should assign the device code (1-5) to the register X22.

A stand-alone system will normally use either a printer or screen as an output device. Programs for either a master or slave station in a B20 cluster will use either a spooler or screen for reports that do not require preprinted forms.

Always use the Printer option for programs using preprinted forms. The report will be directed to the local B20 printer.

Reports which are output to the screen on B21s should be less than 80 columns wide. However, reports which are output to the screen on B22s can use up to 132 columns.

If you select the Available Printer option, the program will first try to print to the spooler. If the spooler is not installed, the program will automatically attempt direct printing.

Be sure to select the Available Printer option only if a spooler is installed or a printer is connected to the B20. If you try to direct print a report from a station which does not have a printer installed, the report will halt. In this case to continue you must either attach a printer or press ACTION/FINISH.

If you choose the Specific Device/File option, you can print to either a device or file.

Required: Yes.

DEVICE/FILE NAME

Entry: 1-20 characters.

Range: Any character.

Default: None.

Function: This entry identifies the device or file to which the report is to be written. When you specify a device, it must be enclosed in square brackets. For example:

```
[LPT]
[PTR]B
[SPLB]
[VID]
```

If you do not specify a volume and directory name with the file name, the access path will default to the path in effect when the file is created by DMRUN.

Required: Yes, but only if the option is selected.

DATE SEPARATOR

Entry: 1 character.

Range: Any character, including a space.

Default: / (slash)

Function: This prompt determines the character that will separate the month, day, and year digits in a Gregorian date.

Required: No. A blank is a valid separator.

TRAILING SIGNS?

Entry:	1 character.
Range:	Y or N
Default:	N
Function:	If you enter N (no), all numeric output items will have leading signs. If you enter Y (yes), all numeric output items will have trailing signs.
Required:	Yes.

FORMS LOAD PROMPT

Entry:	1-48 characters.
Range:	Any characters, including spaces.
Default:	None.
Function:	If you make an entry here, DMRUN will display your entry and pause so that the application user can load special paper or forms in the printer. The program will begin again when the user indicates the forms are loaded by pressing the GO key.
Required:	No.

FORMS UNLOAD PROMPT

Entry:	1-48 characters.

Range: Any characters, including spaces.

Default: None.

Function: If you make an entry here, DMRUN will display your entry and pause after the entire report has been printed so that the application user can remove the special paper or forms used for the report. The program will terminate only when the operator indicates the forms have been removed by pressing the GO key.

Required: No.

(300) REPORT TEXT

This is the only form in the Report Text section of a Report program. It is used to enter all text for output items, such as the report title and column headings.

You must specify any information you want to appear on the report that does not originate from a data file or register.

HEADING OR MESSAGE

Entry: 1-16 characters.

Range: Any characters, including spaces.

Default: None.

Function: This entry defines all constant output items such as the report title and column headings.

Required: No.

(310) REPORT BODY STRUCTURE

This form is the Report Body section of a Report program. In order to print any data from a file or in any way process primary-file records, a detail element should be specified as an element of your report body structure. You can then optionally specify three levels of header and trailer elements. These elements are used for printing titles and totals for groups of detail elements. Specify the elements in order starting with level 1.

You do not need to specify both a header and trailer at each level. A possible structure would consist of a detail element, a level-1 trailer, and a level-2 trailer.

Grand totals for the entire report can be specified separately in the report trailer and are independent of entries on this form.

HEADER (Levels 1, 2, & 3)

Entry:	1 character.
Range:	Y or N
Default:	N
Function:	If you enter Y (yes) for one of these elements, you will be able to specify output items for a group header on a subsequent form.
Required:	Yes.

DETAIL ELEMENT

Entry:	1 character.

Range: Y or N

Default: N

Function: If you enter Y (yes) for this element, you will be able to specify output items for a detail element and/or for detail processing on a subsequent form.

Required: Yes.

TRAILER (Levels 1, 2, & 3)

Entry: 1 character.

Range: Y or N

Default: N

Function: If you enter Y (yes) for one of these elements, you will be able to specify output items for a group trailer on a subsequent form.

Required: Yes.

SOURCE (Levels 1, 2, & 3)

Entry: 1 character and 3 digits.

Range: Any field defined for the files accessed by this report or any program register.

Default: None.

Function: If any headers or trailers are specified in a report body structure, you must also tell DMRUN when these elements are to be printed. Do this by specifying a data field containing a value that changes at group boundaries.

Suppose you are printing transactions sequenced by transaction number and zip code. Also suppose there are several transactions for each zip code. If you wanted to print totals for all the transactions with the same zip code, you would specify a level-1 trailer.

In this case the field which triggers the printing of the level-1 trailer should be the zip code field. Whenever the zip code changes, DMRUN will print a level-1 trailer with all specified totals, and will continue with the next group of transactions.

Required: For each level of header or trailer you specify in the report, you must enter either a SOURCE entry or the following three entries:

1. FILE LETTER
2. POSITION IN RECORD
3. LENGTH OF DATA

FILE LETTER

Entry: 1 character.

Range: A-I

Default: None.

Function: This entry defines a field which will be used to trigger the printing of headers and trailers. This entry has the same function as the preceding SOURCE entry, but can be used when the data location cannot be specified in terms of a source variable.

Required: For each level of header or trailer you specify in the report, you must enter either a SOURCE entry or the following three entries:

1. FILE LETTER
2. POSITION IN RECORD
3. LENGTH OF DATA

POSITION IN RECORD (Levels 1, 2, & 3)

Entry: 1-4 digits.

Range: 1-1017

Default: None.

Function: This entry defines a field which will be used to trigger the printing of headers and trailers. This entry has the same function as the preceding SOURCE entry, but can be used when the data location cannot be specified in terms of a source variable.

Required: For each level of header or trailer you specify in the report, you must enter either a SOURCE entry or the following three entries:

1. FILE LETTER
2. POSITION IN RECORD
3. LENGTH OF DATA

LENGTH OF DATA (Levels 1, 2, & 3)

Entry: 1-2 digits.

Range: 1-45

Default: None.

Function: This entry defines a field which will be used to trigger the printing of headers and trailers. This entry has the same function as the preceding SOURCE entry, but can be used when the data location cannot be specified in terms of a source variable.

Required: For each level of header or trailer you specify in the report, you must enter either a SOURCE entry or the following three entries:

1. FILE LETTER
2. POSITION IN RECORD
3. LENGTH OF DATA

(320) REPORT HEADER OUTPUT ITEMS

This form is in the Report Header/Trailer section of a Report program. A report header is printed only once at the beginning of a report and is an optional report element. No entries are required on this form. It can be used to print identifying information that applies to the entire report.

You can use a report header to print the operator's name, the name of the person requesting the report, or the report options selected for this report. All such information must come from Program Registers, X Registers, or the Report Text table since no application files are read before the report header is processed. The report header can contain up to 100 output items.

A report header page does not have a top margin or a bottom margin. The first line of a page is always line 1, and full access is possible to all lines of a page. A report header is not, however, restricted to one page. If you advance to the top of a page by entering the code 999, you will advance to line 1 of the next page.

SOURCE

Entry:	1 character and 3 digits.
Range:	Any Register or Report Text Identifier.
Default:	None.
Function:	This entry indicates the register or report constant from which DMRUN obtains information for the output item.
Required:	No. An output item without a source can be specified to cause a line advance.

PRINT POSITION

Entry: 1-3 digits.

Range: 1-250

Default: None.

Function: This entry indicates the starting print position of this output item.

Required: No. If a print position is not specified, the output item can be added to a TOTAL REGISTER or referenced in function statements, but it will not be printed.

OUTPUT DESC

Entry: 1-5 characters.

Range: Any valid field description.

Default: The existing description of the output item. These descriptions are:

P1-P40	+10.5
P41-P41	45
T1-T16	+10.5
S1-S16	+10.5
R1-R40	16

Function: In some cases, you may need to specify a field description for output that is different from the default. If you wish to print, for example, only the whole-number portion of an S4.2 field, enter S4.0 for the output description. Be careful not to specify output descriptions unnecessarily because reformatting many numeric output items can reduce report performance.

Note that when reformatting takes place, checks are not made for truncation of significant digits, and numbers are not rounded. Also, when a negative field is reformatted without a sign, a warning is not given.

Required: No.

NUMERIC EDIT

Entry: 1 digit.

Range: 0-7

Default: 0

Function: This entry determines the output editing feature that is applied to numeric output items. The entry is ignored for alphanumeric output items. The meanings of the various output edit selections are explained at the bottom of the entry form.

Required: No.

TOTAL REGISTER

Entry: 1-2 digits.

Range: 1-16

Default: None.

Function: This entry applies only to numeric output items. If you enter 1-16, the value of this output item will be added to the specified total register. Output items can be totalled without being printed. See explanation of PRINT POSITION, above.

Required: No.

LINE ADV

Entry: 1-3 digits.

Range: 1-999

Default: None.

Function: This entry tells DMRUN that the output item is the last item to be printed on a line of the report. The entry also indicates how many lines the printer will be advanced after this line is printed. Enter the value 998 to advance to the page trailer area and print the page trailer. Enter the value 999 to advance to the top of the next page.

Required: No.

(330) REPORT TRAILER OUTPUT ITEMS

This form is in the Report Header/Trailer section of a Report program. A report trailer is printed only once at the end of a report. This trailer is an optional report element. No entries on this form are required.

You can use a report trailer to print totals or summary information that applies to the entire report. All totals referenced in this section will automatically be grand totals. A report trailer can contain up to 100 output items.

A report trailer page does not have a top margin or a bottom margin. The first line of a page is always line 1, and full access is possible to all lines of a page. A report trailer is not, however, restricted to one page. If you advance to the top of a page by entering the code 999, you will advance to line 1 of the next page.

The fields on this form have the same validation requirements as the entries made on (320) Report Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(340) PAGE HEADER OUTPUT ITEMS

This form is found in the Page Header/Trailer section of a Report program. A page header is an optional report element. No entries are required on this form.

A page header is printed once at the top of each page of the report. You can use this header to print a report title, column headings, page number, or other such information. The page header can contain up to 100 output items.

SOURCE

Entry:	1 character and 3 digits.
Range:	Any Register or Report Text Identifier, or any Field Identifier valid for files accessed by this program.
Default:	None.
Function:	This entry indicates the Register, Report Text Identifier, or field from which DMRUN obtains information for the output item.
Required:	No. An output item without a source can be specified to cause a line advance.

PRINT POSITION

Entry:	1-3 digits.
Range:	1-250
Default:	None.

Function: This entry indicates the starting print position of this output item.

Required: No. If a print position is not specified, the output item can be added to a TOTAL REGISTER or referenced in function statements, but it will not be printed.

OUTPUT DESC

Entry: 1-5 characters.

Range: Any valid field description.

Default: The existing description of the output item. These descriptions are:

P1-P40	+10.5
P41-P45	45
T1-T16	+10.5
S1-S16	+10.5
R1-R40	16

Function: In some cases, you may need to specify a field description for output that is different from the default. If you wish to print, for example, only the whole-number portion of an S4.2 field, enter S4.0 for the output description. Be careful not to specify output descriptions unnecessarily because reformatting many numeric output items can reduce report performance.

Note that when reformatting takes place, checks are not made for truncation of significant digits, and numbers are not rounded. Also, when a negative field is reformatted without a sign, a warning is not given.

Required: No.

NUMERIC EDIT

Entry: 1 digit.

Range: 0-7

Default: 0

Function: This entry determines the output editing feature that is applied to numeric output items. The entry is ignored for alphanumeric output items. The meanings of the various output edit selections are explained at the bottom of the entry form.

Required: No.

TOTAL REGISTER

Entry: 1-2 digits.

Range: 1-16

Default: None.

Function: This entry applies only to numeric output items. If you enter 1-16, the value of this output item will be added to the specified total register. Output items can be totalled without being printed. See explanation of PRINT POSITION, above.

Required: No.

LINE ADV

Entry: 1-3 digits.

Range: Blank or 1-999

Default: None.

Function: This entry tells DMRUN that the output item is the last item to be printed on a line of the report. The entry also indicates how many lines the printer will be advanced after this line is printed. Enter the value 998 to advance to the page trailer area and print the page trailer. Enter the value 999 to advance to the top of the next page.

Required: No.

(350) PAGE TRAILER OUTPUT ITEMS

This form is in the Page Header/Trailer section of a Report program. A page trailer is printed only once at the bottom of each page of the report. This trailer is an optional report element. No entries are required on this form.

You can use a page trailer to print page totals, page numbers, or other such information. The page trailer can contain up to 100 output items.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(360) CONTINUATION PAGE HEADER OUTPUT ITEMS

This form is in the Page Header/Trailer section of a Report program. Use the continuation page header when printing invoices and similar preprinted documents. This trailer is an optional report element. No entries are required on this form.

If you specify a continuation page header, it will be printed once at the top of a page. This header is printed only when a complete level-1 group has not been printed on the previous page.

A level-1 group is a series of one or more related detail elements which can be bound by a level-1 header and/or trailer. A header or trailer must be specified to define the group boundaries, but it does not need to contain any output items.

A continuation page header is usually shorter than a standard header. You can use the continuation header for multiple-page, preprinted documents. Print a complete heading on the first page and a continuation header on each of the following pages. The continuation page header can contain up to 100 output items.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(370) CONTINUATION PAGE TRAILER OUTPUT ITEMS

This form is in the Page Header/Trailer section of a Report program. Use the continuation page trailer when printing invoices and similar preprinted forms. This is an optional report element. No entries are required on this form.

The continuation page trailer is usually shorter than a standard trailer and will be printed at the bottom of each page where a level-1 group has not been completely printed. A continuation trailer is always followed by a continuation header when one is defined. A continuation page trailer can contain up to 100 output items.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(380) DETAIL ELEMENT OUTPUT ITEMS

This form is in the Report Body section of a Report program. The Detail Element Output Items are the basic building blocks of the report body structure. Include them if you need to print data and/or accumulate totals from files with your report.

The processing of Detail Element Output Items is driven by records in the primary file. Each record read from the primary file causes DMRUN to process your Detail Element Output Item specifications. Other report elements in the report body are processed as exceptions only when triggered by the fields specified on the (310) Report Body Structure form. A simple file-listing report, for example, would be composed of detail elements only. A detail element can contain up to 100 output items.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(390) LEVEL 1 HEADER OUTPUT ITEMS

This form appears in the Report Body section of a Report program if you request a level-1 header on (310) Report Body Structure. A level-1 header is printed every time there is a change in the contents of any trigger field specified on form 310.

The level-1 header is an optional grouping element. If you want to group your detail elements and provide a heading for each of these groups, specify a level-1 header. It can contain up to 100 output items. If you do not need grouping headers in your report, do not enter information on this form.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results.

(400) LEVEL 1 TRAILER OUTPUT ITEMS

This form appears in the Report Body section of a Report program if you request a level-1 trailer on (310) Report Body Structure. A level-1 trailer is printed every time there is a change in the contents of any trigger field specified on form 310.

The level-1 trailer is an optional grouping element. If you want to group your detail elements and provide trailer information for each of these groups, specify a level-1 trailer. You can enter up to 100 output items. If you do not need grouping trailers in your report, do not enter information on this form.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(410) LEVEL 2 HEADER OUTPUT ITEMS

This form appears in the Report Body section of a Report program if you request a level-2 header on (310) Report Body Structure. A level-2 header is printed every time there is a change in the contents of the level-3 or level-2 trigger fields specified on form 310. The header appears on the report immediately before the first level-1 header in the level-2 group.

The level-2 header is an optional grouping element. If you want a second-level grouping for your detail elements and a heading element at each second-level group boundary, enter your output items for a level-2 header on this form. You can enter up to 100 output items. If you do not need a second-level grouping in your report, do not use this form.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(420) LEVEL 2 TRAILER OUTPUT ITEMS

This form appears in the Report Body section of a Report program if you request a level-2 trailer on (310) Report Body Structure. A level-2 trailer is printed every time there is a change in the contents of the level-3 or level-2 trigger fields specified on form 310. It will appear on the report immediately after the last level-1 trailer in the level-2 group.

The level-2 trailer is an optional grouping element. If you want a second-level grouping for your detail elements and a trailer element at each second-level group boundary, enter your output items for a level-2 trailer on this form. You can enter up to 100 output items. If you do not need a second-level grouping, do not use this form.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(430) LEVEL 3 HEADER OUTPUT ITEMS

This form appears in the Report Body section of a Report program if you request a level-3 header on (310) Report Body Structure. A level-3 header is printed every time there is a change in the contents of the level-3 trigger fields specified on form 310. The header appears on the report immediately before the first level-2 header in the level-3 group.

The level-3 header is an optional grouping element. If you want a third-level grouping in your detail elements and a heading element at each third-level boundary, enter your output items for a level-3 header on this form. You can enter up to 100 output items. If you do not need a third-level grouping in your report, do not use this form.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(440) LEVEL 3 TRAILER OUTPUT ITEMS

This form appears in the Report Body section of a Report program if you request a level-3 trailer on (310) Report Body Structure. A level-3 trailer is printed every time there is a change in the contents of the level-3 trigger fields specified on form 310. It will appear on the report immediately after the last level-2 trailer in the level-3 group.

The level-3 trailer is an optional grouping element. If you want a third-level grouping for your detail elements and a trailer element at each third-level group boundary, enter your output items for a level-3 trailer on this form. You can enter up to 100 output items. If you do not need a third-level grouping in your report, do not use this form.

The fields on this form have the same validation requirements as the entries made on (340) Page Header Output Items. The entries on both forms produce corresponding results. Please refer to the documentation on that form for further information.

(450) CONTROL FILE RECORD SELECTION

This form appears as a part of the Record Selection section of a Report program whenever the control file is not the primary file. You determine whether or not the control file is the primary file when you make entries on (270) Input File Specifications.

The function statements you enter on this form (450) will be executed immediately following each read of the control file. A cancel instruction is used to omit unwanted records. For maximum efficiency, record selection based on information in the control file should be specified on this form. You can enter up to 80 function statements here. If your report does not require this feature, do not enter information on this form.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. Further information can be found in the Function Statements section of this manual. Information on record selection techniques is in the Report Program section.

(460) PRIMARY FILE RECORD SELECTION

This form is in the Record Selection section of a Report program. The function statements that you enter here will be executed immediately following each read of the primary file. A cancel instruction is used to omit unwanted records. For maximum efficiency, record selection based on information in the primary file should be specified on this form. You can enter up to 80 function statements here. If your report does not require this feature, do not make any entries on this form.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. Further information can be found in the Function Statements section of this manual. Information on record selection techniques is in the Report Program section.

(470) SECONDARY FILE RECORD SELECTION

This form appears in the Record Selection section of a Report program if any secondary files are specified on (270) Input File Specifications. The function statements entered on this form will be executed immediately after the records associated with the primary-file record have been read from all secondary files. A cancel instruction is used to omit unwanted records. Record selection based on information in secondary files should be specified on this form. You can enter up to 80 function statements here. If your report does not require this feature, do not make any entries on this form.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. Further information can be found in the function statements section of this manual. Information on record selection techniques is in the Report Program section.

(480) REPORT HEADER FUNCTION

This form is in the Report Header/Trailer section of a Report program. The function statements you enter here will be executed immediately before the report header is printed. Since no files will have been read at this point, reference only Register and Report Constant Identifiers.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. You can enter up to 40 function statements here. Further information can be found in the Function Statements section of this manual.

(490) REPORT TRAILER FUNCTION

This form is in the Report Header/Trailer section of a Report program. The function statements you enter here will be executed immediately before the report trailer is printed. Your function statements can reference only Register, Total, and Report Constant Identifiers.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. You can enter up to 40 function statements here. Further information can be found in the Function Statements section of this manual.

(500) PAGE HEADER FUNCTION

This form is in the Page Header/Trailer section of a Report program. The function statements you enter here will be executed immediately before each page header is printed. Although fields in the first records to be printed on the page are available, your function statements will normally reference Register and Report Constant Identifiers only.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. You can enter up to 40 function statements here. Further information can be found in the Function Statements section of this manual.

(510) PAGE TRAILER FUNCTION

This form is in the Page Header/Trailer section of a Report program. The function statements you enter here will be executed immediately before each page trailer is printed. Although fields in the last record printed on this page are available, your function statements will normally reference Register and Report Constant Identifiers only.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. You can enter up to 40 function statements in this form. Further information can be found in the Function Statements section of this manual.

(520) CONTINUATION PAGE HEADER FUNCTION

This form is in the Page Header/Trailer section of a Report program. The function statements you enter here will be executed immediately before each continuation page header is printed. Although fields in the first records to be printed are available, your function statements will normally reference Register and Report Constant Identifiers only.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. You can enter up to 40 function statements here. Further information can be found in the Function Statements section of this manual.

(530) CONTINUATION PAGE TRAILER FUNCTION

This form is in the Page Header/Trailer section of a Report program. The function statements you enter here will be executed immediately before each continuation page trailer is printed. Although fields in the last record printed on this page are available, your function statements will normally reference Register and Report Constant Identifiers only.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. You can enter up to 40 function statements here. Further information can be found in the Function Statements section of this manual.

(540) DETAIL ELEMENT FUNCTION

This form is in the Report Body section of a Report program. The function statements you enter here will be executed before each detail element is printed.

Entries on this form must use function statement syntax to avoid a run-time failure of your program. You can enter up to 40 function statements here. Further information can be found in the Function Statements section of this manual.

(550) PRIMARY FILE INDICES

This is the only form in the Primary File Keys section of Maintenance, Real-time Update, and Menu programs.

A file can contain up to five different keys. Each key can be either ascending or descending. You can set any key except the first to allow duplicates.

BEGINNING FIELD NUMBER

Entry:	1-3 digits.
Range:	Any field defined in the Field Definitions section.
Default:	None.
Function:	This entry identifies the first field that is part of this key. Note that for Index 1, the beginning field must always be 1. This is entered for you automatically.
Required:	Yes.

ENDING FIELD NUMBER

Entry:	1-3 digits.
Range:	Any field defined in the Field Definitions section.
Default:	None.

Function: This entry identifies the last field that is to be part of this key.

Required: Yes.

ASCENDING?

Entry: 1 character.

Range: Y or N

Default: Y

Function: If you enter Y (yes), this index will be in ascending order.

If you enter N (no), this index will be in descending order.

Required: Yes.

ALLOW DUPLICATES?

Entry: 1 character.

Range: Y or N

Default: N

Function: If you enter N (no), no duplicate keys will be allowed for this index.

If you enter Y (yes), this index can contain duplicate keys.

NOTE

The first index cannot allow duplicates.

Required: Yes.

APPENDIX B

DMCREATE ERROR MESSAGES

The following is a list of B20 Data Manager error messages.

1 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

2 CANNOT OPEN <DM> DMCFORMS.

Reload DMCFORMS from your Data Manager release disks to the <DM> directory.

3 DMCFORMS FILE ERROR.

DMCREATE could not close DMCFORMS.

4 FORM MISSING OR CORRUPT IN DMCFORMS.

DMCREATE could not access a programming form in DMCFORMS. Reload DMCFORMS from your Data Manager release disks.

5 DMCREATE FORM CONTAINS TOO MANY LINES.

A form has been incorrectly modified in DMCFORMS.

6 FORMS RUNTIME ERROR IN UNDISPLAYFORM OR DEFAULTFORM.

Forms run-time has returned an unexpected error. It could be either a system-software or Data Manager problem.

7 FORMS RUNTIME ERROR IN UNDISPLAYFORM OR DEFAULTFORM.

Forms run-time has returned an unexpected error. It could be either a system-software or Data Manager problem.

8 YOUR ENTRY MUST BE ALL NUMBERS.

9 DMCREATE FORM FIELD MISSING.

A form has been incorrectly modified in DMCFORMS.

10 DMCREATE FORM FIELD MISSING.

A form has been incorrectly modified in DMCFORMS.

11 DMCCREATE FORM FIELD MISSING.

A form has been incorrectly modified in DMCFORMS.

12 B20 SYSTEM ERROR.

Beep (to sound alarm) returned an error. This is a system-software problem.

13 B20 SYSTEM ERROR.

PutFrameChars (blank out error message) returned an error. This is a system-software problem.

14 YOUR ENTRY MAY NOT BE BLANK.

15 YOUR DESCRIPTION MUST BE AN ALPHA DESCRIPTION.

16 YOUR ENTRY MAY NOT BE BLANK.

17 YOUR DESCRIPTION MUST BE AN ALPHA DESCRIPTION.

18 CANNOT OPEN <DM> DMCDATA.

Note that this error message cannot be translated.

DAM error when opening DMCDATA. Reload DMCDATA from your Data Manager release disks to the <DM> directory.

19 DMCDATA FILE ERROR.

DMCREATE could not close DMCDATA.

20 DMCDATA FILE ERROR.

DMCDATA record not found. The file is corrupt. Reload DMCDATA from your release disk.

21 DMCDATA FILE ERROR.

DMCDATA read error. The file is corrupt, or a system-software problem has occurred.

22 YOUR ENTRY MAY NOT BE BLANK.

23 YOUR ENTRY MUST BE "Y" OR "N".

24 DM APPLICATION PROGRAM CORRUPT.

Invalid program type in existing program. Program is unusable.

25 DM APPLICATION PROGRAM CORRUPT.

Invalid program type in existing program. Program is unusable.

26 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

27 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

28 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

29 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

30 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

31 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

32 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

33 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

34 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

35 YOUR SELECTION IS NOT AVAILABLE. PLEASE REENTER.

36 YOUR ENTRY MAY NOT BE BLANK.

Options are: enter 0, press the F10 key to delete range, or press the GO key to move to next form.

37 YOUR ENTRY CONTAINS AN ILLEGAL OR MISPLACED CHARACTER.

The only non-numeric character allowed before the decimal point is a sign.

38 YOUR ENTRY CONTAINS AN ILLEGAL OR MISPLACED CHARACTER.

Non-numeric characters are not allowed after the decimal point.

39 NUMERIC ENTRIES MUST BE 15 DIGITS OR LESS.

40 UNABLE TO DISPLAY FIELD DEFINITION. PROGRAM CORRUPT.

41 UNABLE TO DISPLAY FIELD DEFINITION. PROGRAM CORRUPT.

42 UNABLE TO DISPLAY FIELD DEFINITION. PROGRAM CORRUPT.

43 UNABLE TO DISPLAY FIELD DEFINITION. PROGRAM CORRUPT.

44 UNABLE TO DISPLAY FIELD DEFINITION. PROGRAM CORRUPT.

45 UNABLE TO DISPLAY FIELD DEFINITION. PROGRAM CORRUPT.

46 NEXT FIELD MUST BE GREATER THAN 0, LESS THAN LAST FIELD + 2.

47 STD DESCRIPTION MUST BE 1-28.

48 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

49 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

50 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

51 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

52 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

53 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

54 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

55 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

56 B20 SYSTEM ERROR.

Error returned while decoding valid 8-bit numerics. This is a system-software error.

57 B20 SYSTEM ERROR.

Error returned while decoding valid 8-bit numerics. This is a system-software error.

58 DATA MANAGER MAXIMUM ALPHA FIELD SIZE IS 45 CHARACTERS.

59 DATA MANAGER MAXIMUM NUMERIC FIELD SIZE IS 15 DIGITS.

60 ALL FIELD DEFINITIONS MUST CONTAIN A DESCRIPTION.

61 FIELD NUMBERS MUST BE 1-100.

62 PROGRAM REGISTER NUMBERS MUST BE 1-45.

63 "X" REGISTER NUMBERS MUST BE 1-23.

64 SOURCE MAY REFERENCE "A"-"I", "P", OR "X".

65 NUMBER IN SOURCE MAY NOT CONTAIN SPACES OR ALPHA CHARACTERS.

Error caused by entering a source number containing a blank or alpha character. For example, A2 3, A2R.

66 SOURCE MUST CONTAIN FIELD OR REGISTER NUMBER.

67 SKIP TO FIELD MUST BE > NUMBER OF FIELD BEING DEFINED AND < 100.

68 YOUR ENTRY MUST BE A WHOLE NUMBER.

69 RANGE MUST BE 1-20.

70 UNABLE TO DISPLAY FIELD DEFINITION. PROGRAM CORRUPT.

71 ACCESS CONTROL MUST BE 1-4.

72 ACCESS CONTROL MUST BE 1-2.

73 INDEX NUMBER MUST BE 1-5.

74 YOUR ENTRY MAY NOT BE BLANK.

75 FIELD NOT DEFINED IN PROGRAM.

A field referred to as part of a primary-file key was not defined as a field in this program.

76 PRINT POSITION MUST BE 0-250.

77 INCORRECT DESCRIPTION. SEE HELP FORM FOR EXAMPLES.

See (150) Field Definition form in Appendix A for an explanation of Descriptions.

78 YOUR ENTRY MAY NOT BE BLANK.

Your options are to enter a range or press the GO key to continue to the next form.

79 TABLE IS FULL. CANNOT INSERT AN ENTRY.

80 "N" NOT ALLOWED UNTIL ALL DETAIL ITEMS/STATEMENTS ARE DELETED.

81 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

82 SOURCE MUST CONTAIN FIELD OR REGISTER NUMBER.

83 YOUR DESCRIPTION MUST BE A NUMERIC DESCRIPTION.

84 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

85 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

86 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

87 "CONTROL" IS NOT PERMITTED AS A PROGRAM NAME.

88 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

89 ISAM NOT INSTALLED. INSTALL ISAM BEFORE CONTINUING.

90 INVALID PRIMARY KEY SPECIFICATION.

The ending field number of a primary-file key is less than the beginning field number.

91 B20 SYSTEM ERROR.

Cannot open print file.

92 SECONDARY KEY FIELD SPECIFIED NOT DEFINED IN PROGRAM.

A field referenced as a key field to a secondary file was not defined as a field in this program.

93 SECONDARY KEY FIELD SPECIFIED NOT DEFINED IN PROGRAM.

A field referenced as a key field to a secondary file was not defined as a field in this program.

94 YOUR DESCRIPTION MUST BE A NUMERIC DESCRIPTION.

95 DATA MANAGER MAXIMUM NUMERIC FIELD SIZE IS 15 DIGITS.

96 INVALID PRIMARY KEY SPECIFICATION.

One of your primary-file keys includes a field(s) not defined in this program.

97 PRIMARY KEY IS GREATER THAN 64 BYTES. REENTER PRIMARY KEY FIELDS.

98 STANDARD DESCRIPTIONS 26-28 MUST ALSO HAVE A DESCRIPTION.

Standard Descriptions 26-28 refer only to the way the field is stored in the record (integer, single-precision real number, or double-precision real number). You must enter a description of the display format.

99 FIELD NOT DEFINED IN PROGRAM.

A field referenced as a key field to a secondary file was not defined as a field in this program.

100 FIELD NOT DEFINED IN PROGRAM.

A field referenced as a key field to a secondary file was not defined as a field in this program.

101 POSITION NOT WITHIN RECORD DEFINED BY PROGRAM.

102 POSITION NOT WITHIN RECORD DEFINED BY PROGRAM.

103 POSITION NOT WITHIN RECORD DEFINED BY PROGRAM.

104 PARTIAL FIELDS MAY NOT OVERLAP FIELD BOUNDARIES.

When entering position and length, the field described must not cross field boundaries established when the record was defined. Use several positions and lengths or combine field-number entries with positions and lengths to define the key.

105 FIELD LENGTH MUST BE 1-45.

106 FIELD LENGTH MUST BE 1-45

107 YOUR ENTRY MAY NOT BE BLANK.

Your options are to enter a statement or press the GO key to continue to the next form.

108 TYPE MUST BE 1-8.

109 YOUR DESCRIPTION MUST BE A NUMERIC DESCRIPTION.

110 TYPE MUST BE 1-8.

111 DATA MANAGER MAXIMUM INTEGER SIZE IS 4 DIGITS.

112 CONTROL FILE MAY REFERENCE "A"-"I".

113 DATA MANAGER MAXIMUM NUMERIC FIELD SIZE IS 15 DIGITS.

114 TABLE IS FULL. CANNOT INSERT AN ENTRY.

115 TABLE IS FULL. CANNOT INSERT AN ENTRY.

116 TABLE IS FULL. CANNOT INSERT AN ENTRY.

117 TABLE IS FULL. CANNOT INSERT AN ENTRY.

118 TABLE IS FULL. CANNOT INSERT AN ENTRY.

119 SECONDARY KEY FIELD SPECIFIED NOT DEFINED IN PROGRAM.

A field referenced as a key field to a secondary file was not defined as a field in this program.

120 TABLE IS FULL. CANNOT INSERT AN ENTRY.

121 TABLE IS FULL. CANNOT INSERT AN ENTRY.

122 TABLE IS FULL. CANNOT INSERT AN ENTRY.

123 TABLE IS FULL. CANNOT INSERT AN ENTRY.

124 PAGE SIZE MUST BE 1-255.

125 MARGIN MUST BE LESS THAN PAGE SIZE.

126 BOTTOM MARGIN MUST BE LESS THAN PAGE SIZE - TOP MARGIN.

127 OUTPUT DEVICE MUST BE 1-5.

128 TABLE IS FULL. CANNOT INSERT AN ENTRY.

129 YOUR ENTRY MAY NOT BE BLANK.

Your options are to enter a heading or press the GO key to continue to the next form.

130 DM APPLICATION PROGRAM CORRUPT.

Source for output item cannot be encoded into ASCII.

131 TABLE IS FULL. CANNOT INSERT AN ENTRY.

132 TABLE IS FULL. CANNOT INSERT AN ENTRY.

133 TABLE IS FULL. CANNOT INSERT AN ENTRY.

134 POSITION NOT WITHIN RECORD DEFINED BY PROGRAM.

135 <DM> DMCHELP NOT AVAILABLE.

136 FIELD NUMBERS MUST BE 1-100.

137 TEXT NUMBER NOT DEFINED IN REPORT TEXT SECTION.

138 PROGRAM REGISTER NUMBERS MUST BE 1-45.

139 "X" REGISTER NUMBERS MUST BE 1-23.

140 TOTAL REGISTER NUMBER MUST BE 1-16.

141 SOURCE MAY REFERENCE "A"-"I", "P", "R", "S", "T", OR "X".

142 NUMBER IN SOURCE MAY NOT CONTAIN SPACES OR ALPHA CHARACTERS.

Error caused by entering a source number containing a blank or alpha character. For example, A2 3, A2R.

143 SOURCE MUST CONTAIN FIELD OR REGISTER NUMBER.

144 B20 SYSTEM ERROR.

Error in UndisplayForm when attempting to display Form Help.

145 B20 SYSTEM ERROR.

Unable to open a file for return from chained program.

146 B20 SYSTEM ERROR.

Unable to open file for return from chained program.

151 LIST PROGRAM (<DM> DMPLIST) NOT FOUND.

152 RUN PROGRAM (<DM> DMRUN) NOT FOUND.

153 CANNOT PRINT DIRECT. (PRESS "GO" TO SPOOL, "FINISH" TO END.)

154 NUMERIC EDIT SELECTION MUST BE 0-7.

155 TOTAL REGISTER NUMBER MUST BE 1-16.

156 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

157 DM APPLICATION PROGRAM CORRUPT.

Source for control breaks cannot be encoded into ASCII.

158 B20 SYSTEM ERROR.

Cannot close print file.

159 ENTRY CAN'T BE CHANGED UNTIL OUTPUT ITEMS IN ELEMENT ARE DELETED.

160 FIELD NUMBERS MUST BE 1-100.

161 PROGRAM REGISTER NUMBERS MUST BE 1-45.

162 SOURCE MAY REFERENCE "A"-"I" OR "P".

163 NUMBER IN SOURCE MAY NOT CONTAIN SPACES OR ALPHA CHARACTERS.

Error caused by entering a source number containing a blank or alpha character. For example, A2 3, A2R.

164 SOURCE MUST CONTAIN FIELD OR REGISTER NUMBER.

165 FILE LETTER MAY REFERENCE "A"-"I".

166 POSITION NOT WITHIN RECORD DEFINED BY PROGRAM.

167 FIELD LENGTH MUST BE 1-45.

168 PROGRAM NOT IN LIBRARY.

169 YOUR ENTRY MAY NOT BE BLANK.

170 DMCDATA FILE ERROR.

Records missing from DMCDATA. Reload DMCDATA from your release disk to the <DM> directory.

171 B20 SYSTEM ERROR.

Error in QUERYFRAMECHAR to obtain directory image from the screen.

172 B20 SYSTEM ERROR.

Error in WRITEBSRECORD when attempting to print directory.

173 B20 SYSTEM ERROR.

Error in WRITEBSRECORD when attempting to print directory.

174 PROGRAM LIBRARY ISAM ERROR.

175 PROGRAM LIBRARY ISAM ERROR.

176 PROGRAM LIBRARY ISAM ERROR.

177 PROGRAM LIBRARY ISAM ERROR.

178 PROGRAM LIBRARY ISAM ERROR.

179 PROGRAM LIBRARY ISAM ERROR.

180 DUPLICATE "COPY TO" PROGRAM NAME.

181 PROGRAM LIBRARY ISAM ERROR.

182 PROGRAM LIBRARY ISAM ERROR.

183 CANNOT ALLOCATE MEMORY.

AllocMemorySL failure. Check error code for specific problem.

184 DATA MANAGER SYSTEM ERROR.

DM memory-allocation stack has overflowed.

185 B20 SYSTEM ERROR.

DeallocMemorySL failure. Check error code for specific problem.

186 DATA MANAGER SYSTEM ERROR.

DM attempted to de-allocate memory that had not been allocated.

187 PROGRAM LIBRARY ISAM ERROR.

Error in OpenIsam. Error code can apply to either data file or index file.

188 PROGRAM LIBRARY ISAM ERROR.

Error in CloseIsam.

189 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to load program table from Program Library into memory.

190 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to load program table from Program Library into memory.

191 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to unload program table from memory to Program Library.

192 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to unload program table from memory to Program Library.

193 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to unload program table from memory to Program Library.

194 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to unload program table from memory to Program Library.

195 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to unload program table from memory to Program Library.

196 DATA MANAGER SYSTEM ERROR.

DMCREATE tried to access a nonexistent LED.

197 KEYBOARD LED NOT WORKING. WILL NOT AFFECT PROGRAMMING.

198 PROGRAM NOT IN LIBRARY.

199 DMCDATA FILE ERROR.

Records missing from DMCDATA. Reload DMCDATA from your release disk to the <DM> directory.

200 B20 SYSTEM ERROR.

ResetFrame failed when attempting to display help form.

201 PROGRAM INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

202 BOTH RANGE LIMITS CANNOT BE BLANK.

203 ENTER EITHER A SOURCE OR A LINE ADVANCE.

204 SELECTION FOR CONTROL FILE MUST HAVE CORRESPONDING PROGRAM NAME.

You cannot select a file as your control file unless you have already entered a program name for it in the fields above.

205 HELP FORM MISSING FROM DMCHELP.

Field help form missing from DMCHELP.

206 DMCREATE FORM CONTAINS TOO MANY LINES.

A form has been incorrectly modified in DMCHELP.

207 HELP FORM MISSING FROM DMCHELP.

Form Help form missing from DMCHELP.

208 DMCREATE FORM CONTAINS TOO MANY LINES.

A form has been incorrectly modified in DMCHELP.

209 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

210 ENTRY MUST BE BLANK UNLESS ELEMENT IS SPECIFIED ABOVE.

211 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

212 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

213 ENTRY MUST BE BLANK UNLESS ELEMENT IS SPECIFIED ABOVE.

214 PROGRAM LIBRARY INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

215 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to rewrite Program Library control record.

216 ENTRY MUST BE BLANK UNLESS ELEMENT IS SPECIFIED ABOVE.

217 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to write Program Library control record.

218 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to read Program Library control record.

219 ENTRY MUST BE BLANK UNLESS ELEMENT IS SPECIFIED ABOVE.

220 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to open a new Program Library.

221 PROGRAM LIBRARY ISAM ERROR.

ISAM error while attempting to create a new Program Library.

222 CANNOT READ CONTROL RECORD IN DMCDATA.

Note that this error message cannot be translated.

Reload DMCDATA from your release disk to the <DM> directory.

223 DMCDATA INCOMPATIBLE WITH THIS VERSION OF DMCREATE.

Note that this error message cannot be translated.

Reload DMCDATA from your release disk to the <DM> directory.

224 B20 SYSTEM ERROR.

VAM error while writing error message.

225 DATA MANAGER SYSTEM ERROR.

DM unable to allocate memory to store programming form while attempting to display help form.

226 B20 SYSTEM ERROR.

Unable to restore programming form after help form is displayed.

227 B20 SYSTEM ERROR.

Unable to de-allocate memory after restoring programming form.

228 FIELDS WITH STD DESC 26-28 CANNOT BE PART OF A PRIMARY KEY.

APPENDIX C

DMRUN ERROR MESSAGES

The following is a list of B20 Data Manager error messages.

1 PROGRAM LIBRARY FILE ERROR.

The specified program library has no control record. Make sure that the name specified for the library is correct or that the file in question is in fact a Data Manager program library. Correct the name or replace the library file before retrying.

2 BASIC PROGRAM NOT FOUND.

BASIC program was not found. Make sure that the name specified for this program in the link table of the previous Data Manager program is correct or that the program in question is present and is in fact a BASIC program. Correct the name or supply the program run file before retrying.

3 B20 SYSTEM ERROR.

B20 System error while chaining a BASIC program. First try initializing the B20. If the error still occurs, try reloading the B20 system software. The operating system error code may provide an indication as to the nature of the problem.

4 B20 SYSTEM ERROR.

B20 System error while chaining a BASIC program. First try initializing the B20. If the error still occurs, try reloading the B20 system software. The operating system error code may provide an indication as to the nature of the problem.

5 BASIC.RUN NOT INITIATED.

The BASIC software could not be initiated while chaining a BASIC program. Make sure that all necessary BASIC software is present. The operating system error code may provide an indication as to the nature of the problem.

6 DMRDATA FILE ERROR.

A write error occurred while rewriting a record in DMRDATA. This is probably due to a hardware disk error. The operating system error code may provide an indication as to the nature of the problem.

7 DMRDATA FILE ERROR.

B20 System error while attempting to close DMRDATA. First try initializing the B20. If on a subsequent run the error persists, try reloading the B20 system software. The operating system error code may provide an indication as to the nature of the problem.

8 DMRDATA FILE ERROR.

DMRDATA cannot be opened. Make sure that DMRDATA is present and is in the DM directory.

9 ADDITIONAL MEMORY REQUIRED.

Insufficient memory. The host B20 must have a minimum of 256KB of memory. Make sure that memory is not being wasted by an unnecessary installation of the spooler or an unnecessarily large ISAM. The operating system error code may provide additional information.

10 DM MEMORY MANAGEMENT ERROR.

A problem exists in the Data Manager interpreter (DMRUN) that affects the management of memory. Please file a trouble report with your software supplier.

11 DM MEMORY MANAGEMENT ERROR.

A problem exists in the Data Manager interpreter (DMRUN) that affects the management of memory. Please file a trouble report with your software supplier. The operating system error code may provide an indication of the nature of the problem.

12 DM MEMORY MANAGEMENT ERROR.

A problem exists in the Data Manager interpreter (DMRUN) that affects the management of memory. Please file a trouble report with your software supplier.

13 DMRDATA FILE ERROR.

A record in DMRDATA, needed by DMRUN, was missing. This is a Data Manager error and should be reported to your software supplier.

14 DMRDATA FILE ERROR.

DMRUN was unable to read a record it needed in DMRDATA. The operating system error code may provide an indication of the nature of the problem. Try replacing the DMRDATA file with a backup copy. It is also possible that a disk hardware problem may exist. Check with your field engineer.

15 DMRDATA FILE ERROR.

DMRDATA is empty or is corrupt. Replace it with a backup copy.

16 YOUR SELECTION IS NOT AVAILABLE. PLEASE ENTER ANOTHER.

The number you have entered does not correspond to an installed application. Please reenter a valid number.

17 DMRDATA FILE ERROR.

An application description record in DMRDATA, needed by DMRUN, was missing. This is a Data Manager error and should be reported to your software supplier.

18 LIBRARY ID/PROGRAM NAME IS REQUIRED.

The program name is a required entry for program type 1 (Data Manager program) or 7 (ISAM Reorganize).

19 INVALID SYSTEM DATE CODE IN SYSTEM FILE CONTROL INFORMATION.

An invalid date code is in a DMRDATA control record. This number must be in the range, 1 - 3. Use DMTRAN to reenter the correct number.

20 LIBRARY ID/PROGRAM NAME IS REQUIRED.

The program name is a required entry for a program type 2, 3, 4, 5, 6, or 8 (i.e., for an external program or submit file).

21 DM ENTRY FORM ERROR.

A non-numeric character has been entered as a selection type. Please reenter a valid number from 1 - 8.

22 THIS ENTRY MUST BE ALL NUMBERS.

The program type code must be a single-digit number in the range 1-8. Reenter a valid program type code.

23 DM ENTRY FORM ERROR.

The direct selection form is corrupt. Try reloading the system forms file, DMRFORMS. Otherwise, try reloading DMRUN. The operating system error code may provide an indication of the nature of the problem.

24 DM ENTRY FORM ERROR.

The direct selection form is corrupt. Try reloading the system forms file, DMRFORMS. Otherwise, try reloading DMRUN. The operating system error code may provide an indication of the nature of the problem.

25 DM ENTRY FORM ERROR.

The direct selection form is corrupt. Try reloading the system forms file, DMRFORMS. Otherwise, try reloading DMRUN. The operating system error code may provide an indication of the nature of the problem.

26 DM ENTRY FORM ERROR.

The direct selection form is corrupt. Try reloading the system forms file, DMRFORMS. Otherwise, try reloading DMRUN.

27 THIS ENTRY MUST BE ALL NUMBERS.

A non-numeric entry has been made as a selection. Please reenter a valid number from 1 - 12.

28 DM ENTRY FORM ERROR.

The direct selection form is corrupt. Try reloading the system forms file, DMRFORMS. Otherwise, try reloading DMRUN. The operating system error code may provide an indication of the nature of the problem.

29 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

33 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

34 DM MEMORY MANAGEMENT ERROR.

A problem exists in the Data Manager interpreter (DMRUN) that affects the management of memory. Please file a trouble report with your software supplier.

35 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

An invalid variable has been declared. A variable name must consist of a single character code, followed immediately be a number. For example: X16, A2, P1, P01.

36 B20 SYSTEM ERROR.

A forms run-time software error has occurred. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

37 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can successfully run.

38 FORM NAME IN PROGRAM NOT FOUND IN FORMS LIBRARY.

A help form has not been found in the application forms library. The form may not have been entered into the library or the form name might be wrong. Make the appropriate correction before requesting help at this point again.

39 PROGRAM HELP FORM ERROR.

Help form cannot be displayed. The operating system error code may provide an indication of the nature of the problem. The form will probably have to be re-created. Do not request help at this entry point until the form can be corrected.

40 B20 SYSTEM ERROR.

A forms run-time software error has occurred. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

41 MORE THAN 20 FIELDS DECLARED FOR A STANDARD FORM PROGRAM.

Attempt to display a field with an invalid field number (0 or greater than 20) on a standard form which only allows a maximum of twenty fields. This indicates a Data Manager failure and should be reported to your software supplier.

42 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

43 <field number> FIELD: FIELD NAME IN PROGRAM NOT FOUND ON FORM.

A forms error has occurred. The most likely cause is a reference to a field name that was not declared on the form. Make sure that the field name in the indicated field corresponds to a field on the form. The operating system error code may provide additional information.

44 FIELD SIZE MISMATCH BETWEEN PROGRAM AND FORMS EDITOR FORM.

The corresponding forms field is too small for the indicated numeric field. Make sure that the form field is large enough to hold all the required digits together with any sign and decimal point.

45 PROGRAM ENTRY FORM ERROR.

A forms run-time software error has occurred. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the problem.

46 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

An invalid source field number has been declared for the indicated field. Correct the source designation of this field before attempting to run this program again.

47 INVALID SYSTEM DATE CODE IN SYSTEM FILE CONTROL INFORMATION.

The control record of the Data Manager system file (DMRDATA) contains an invalid date code. This code should be 1, 2, or 3. (1 represents format MMDDYY; 2 represents format DDMMYY, and 3 represents format YYMMDD).

49 PROGRAM FORM LIBRARY FILE ERROR.

A system error has occurred while attempting to close a form file. Try initializing the B20. The operating system error code may provide an indication of the nature of the problem.

50 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

51 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

52 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

53 FORM NAME IN PROGRAM NOT FOUND IN FORMS LIBRARY.

A forms error has occurred. The most likely cause is a reference to a form name that does not exist in the form file. Make sure that the form has been created and inserted in the form file with the Librarian or that the correct form name has been referenced in your program. The operating system error code may provide additional information.

54 PROGRAM ENTRY FORM ERROR.

A system error has occurred while attempting to display a form. Try initializing the B20. The operating system error code may provide an indication of the nature of the problem.

55 FIELD SIZE MISMATCH BETWEEN PROGRAM AND FORMS EDITOR FORM.

The length of data entered on the screen is larger than provided for according to the description associated with this field. Either reduce the size of the field on the form or alter the description in your program accordingly.

57 THIS ENTRY MUST BE ALL NUMBERS.

Non-numeric character, other than a sign or decimal point, is not allowed in a numeric field. Reenter using a plus sign (+), minus sign (-), or 0-9 only.

58 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

59 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

60 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

61 <field number> FIELD: FIELD NAME IN PROGRAM NOT FOUND ON FORM.

A forms error has occurred. The most likely cause is a reference to a field name that was not declared on the form. Make sure that the field name in the indicated field does correspond to a field on the form. The operating system error code may provide additional information.

62 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

63 PROGRAM ENTRY FORM ERROR.

A forms run-time software error has occurred. Try initializing the B20 or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

64 CANNOT OPEN PROGRAM FORMS LIBRARY.

The program forms library cannot be opened. The most likely reason is that either the file is not present or the name declared in the program library control record does not match the actual name of the file. Correct the problem before trying to run any program from this program library.

65 ENTRIES OPEN/CLOSE ERROR.

The indicated file cannot be closed due to an ISAM error. Try initializing or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

66 DM FILE MANAGEMENT ERROR.

A Data Manager error has occurred resulting in an attempt to close a file that was already closed. Your software supplier should be notified of the problem.

67 ENTRIES DELETE ERROR.

A record could not be deleted from the indicated file due to an ISAM error. Try initializing or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

68 FILE MANAGEMENT ERROR IN DM.

A Data Manager error has occurred resulting in an attempt to delete a record from a file that was closed. Your software supplier should be notified of the problem.

69 ENTRIES CREATE ERROR.

An ISAM error has occurred while attempting to create a new data file. The operating system error code may provide an indication of the nature of the problem.

70 ENTRIES NOT FOUND.

The primary file was not present on the disk and this program is not permitted to create a new file. If the file is present, make sure that the file name declared in the program matches the actual file name and that the volume and directory names are also correct.

71 ENTRIES OPEN/CLOSE ERROR.

The indicated file cannot be opened due to an ISAM error. This could be because the record length according to the program does not match the length of the actual record. If the program field specifications have been altered so that the record length is no longer the same as in the file when it was first created, then the old file will have to be removed. If this is not the problem, then it may be a system software problem. Try initializing or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

72 FILE MANAGEMENT ERROR IN DM.

A Data Manager error has occurred resulting in an attempt to open a file that was already open. Your software supplier should be notified of the problem.

73 ENTRIES READ/WRITE ERROR.

An ISAM read error occurred while attempting to read a record from the indicated file. The operating system error code may provide an indication of the nature of the problem.

74 FILE MANAGEMENT ERROR IN DM.

A Data Manager error has resulted in the indicated file being opened for output only so that a record could not be read. Your software supplier should be notified of the problem.

75 ENTRIES READ/WRITE ERROR.

An ISAM read error occurred while attempting to read a record from the indicated file. The operating system error code may provide an indication of the nature of the problem.

76 FILE MANAGEMENT ERROR IN DM.

A Data Manager error has resulted in the indicated file being opened for output only so that a record could not be read. Your software supplier should be notified of the problem.

77 ENTRIES READ/WRITE ERROR.

An ISAM read error occurred while attempting to write a record in the indicated file. The operating system error code may provide an indication of the nature of the problem.

78 FILE MANAGEMENT ERROR IN DM.

A Data Manager error has resulted in the indicated file being opened for input only so that a record could not be written. Your software supplier should be notified of the problem.

79 ENTRIES INDEX ERROR.

An ISAM error has occurred while attempting to set up a file range. This could be the result of the file being corrupt. Try reloading the system software. The operating system error code may provide an indication of the nature of the problem.

80 FILE MANAGEMENT ERROR IN DM.

A Data Manager error has occurred resulting in an attempt to set up a file range in a file that was closed. Your software supplier should be notified of the problem.

81 ENTRIES INDEX ERROR.

An ISAM error has occurred while attempting to set up a file range. This could be the result of the file being corrupt. Try recovering from an earlier copy of the file. Also try reloading the system software. The operating system error code may provide an indication of the nature of the problem.

82 FILE MANAGEMENT ERROR IN DM.

A Data Manager error has occurred, resulting in an attempt to set up a file range in a file that was closed. Your software supplier should be notified of the problem.

83 ENTRIES READ/WRITE ERROR.

An ISAM write error occurred while attempting to write a record in the indicated file. The operating system error code may provide an indication of the nature of the problem.

84 FILE MANAGEMENT ERROR IN DM.

A Data Manager error has resulted in the indicated file being opened for input only so that a record could not be written. Your software supplier should be notified of the problem.

85 ENTRIES MAY NOT BE CHANGED WITH THIS PROGRAM.

A record containing this key already exists. Added records must have a unique key. If it is necessary to be able to change or delete existing records, then the program should be changed to allow change and/or delete functions.

86 END OF NEXT PAGE SEQUENCE. ENTER AN IDENTIFIER TO RESTART.

The end of the primary file has been reached using the NEXT PAGE key. For any further access to the file a specific key must be entered.

87 NO ENTRY STORED WITH THIS IDENTIFIER. TRY ANOTHER IDENTIFIER.

A record with this key does not exist and this program has specified that no new records are to be added.

88 <field number> FIELD: NAME REQUIRED FOR INQUIRY FIELDS.

The specified field has no name and cannot therefore be displayed. This is not allowed in an inquiry program.

89 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

The specified field has no source designation. This is required for inquiry programs.

90 END OF NEXT PAGE SEQUENCE. ENTER AN IDENTIFIER TO RESTART.

The end of the primary file has been reached using the NEXT PAGE key. For any further access to the file a specific key must be entered.

91 NO ENTRY STORED WITH THIS IDENTIFIER. TRY ANOTHER IDENTIFIER.

There is no record in the primary file with this key.

92 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

A Data Manager error has allowed or caused an invalid selection to be requested. Your software supplier should be notified of the problem.

93 <file letter> FILE: ERROR IN FIELD OFFSET AND/OR LENGTH.

The secondary-file key field specification for this file includes a field whose offset and length exceed the bounds of the primary-file record.

94 <file letter> FILE: KEY LENGTH LESS THAN LENGTH OF KEY FIELDS IN PROGRAM.

The secondary-file key field specifications for this file define a key length which is longer than the actual key length for this file.

95 PROGRAM LIBRARY FILE ERROR.

An ISAM error has occurred while attempting to close the program library. This is a B20 system error. Try reloading the system software or replacing the library file itself. The operating system error code may provide an indication of the nature of the problem.

96 PROGRAM LIBRARY INDEX ERROR.

An ISAM error has occurred while attempting to load a Data Manager program. This could be because of a corrupt program library file or a B20 system software problem. Try replacing the library and/or system software. The operating system error code may provide an indication of the nature of the problem.

97 PROGRAM LIBRARY FILE ERROR.

An ISAM read error occurred while attempting to read a record from the program library. The operating system error code may provide an indication as to the nature of the problem.

98 DM PROGRAM NOT FOUND OR INCOMPLETE.

An attempt has been made to load a nonexistent or incomplete program. Check the directory to make sure that the program is present or make sure that the name of the requested program is correct.

99 ADDITIONAL MEMORY REQUIRED TO CONTINUE THIS PROGRAM.

There is insufficient memory available to load this program. Either create a simpler program with lower memory requirements or obtain additional memory for the B20.

100 DM PROGRAM CORRUPT: PROGRAM UNUSABLE.

The program being loaded has an invalid program type code. This could be caused by a corrupt program library or a Data Manager error. Try replacing the program library with a backup copy.

101 DM PROGRAM CORRUPT: PROGRAM UNUSABLE.

The program being loaded has an invalid program type code. This could be caused by a corrupt program library or a Data Manager error. Try replacing the program library with a backup copy.

102 PROGRAM LIBRARY FILE ERROR.

An ISAM error has occurred while attempting to open the program library file. Make sure that the designated file is in fact a program library file. Try replacing the file or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

103 PROGRAM LIBRARY FILE ERROR.

An ISAM read error has occurred while reading the program library file. Try replacing the program library file with a backup copy. This could also be caused by a disk hardware failure. The operating system error code may provide an indication of the nature of the problem.

104 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

No program linkage has been specified for the selection you have made. If the menu form is still on the screen, reenter a valid selection. If DMRUN has returned to a previous menu, then the selection was totally invalid.

105 <file letter> FILE: KEY LENGTH LESS THAN LENGTH OF KEY FIELDS IN PROGRAM.

The primary-file key length as indicated in this program does not match the file's actual key length. Correct the program specifications accordingly. A possible problem could be that the key fields specified were greater than 64 bytes and so the program shows a key length of zero.

108 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

The indicated field has as its source an invalid source designation.

110 <field number> FIELD: REFERENCE TO A RANGE NOT IN RANGE TABLE.

The indicated field specification references a range which is not defined. Correct the program before running it again.

111 ENTRY OUTSIDE NUMERIC RANGE LIMITS. TRY LARGER OR SMALLER.

Your entry is less than the specified range allows. Reenter an acceptable value.

112 THIS ENTRY IS OUTSIDE THE ALPHABETIC RANGE LIMITS.

Your entry is outside the specified range. Reenter an acceptable value.

113 <field number> FIELD: VALIDATION ERROR. RE-ENTER OR CANCEL.

The value obtained for this field does not satisfy the editing requirements for this field and since the field cannot be entered, it cannot be corrected directly. Reenter the record if this can result in the value being corrected, otherwise cancel the record.

114 REFERENCE DOES NOT VERIFY THIS ENTRY. MAKE DIFFERENT ENTRY.

The data you have entered contains a key which does not exist in the indicated secondary file. Reenter the record with correct data or cancel the record.

115 THIS ENTRY MAY NOT BE BLANK. ENTER AT LEAST ONE LETTER.

This field is mandatory. It cannot be left blank. Reenter a non-blank value.

116 <file letter> FILE: ERROR IN FIELD OFFSET AND/OR LENGTH.

A Data Manager error has resulted in the primary file having a record description containing an offset and length that exceed the bounds of its record size. Your software supplier should be notified of the problem.

117 <number> FIELD/STATEMENT/ITEM: SOURCE NOT RECOGNIZED.

The indicated report field specification has a source whose source number is invalid for that source. Correct the report program before running it again.

118 <report item number> ITEM: INVALID "X" REGISTER.

An attempt has been made to access an invalid X Register. The valid X Registers are Xl - X23. You must corrected the program before it can be run successfully.

119 <number> FIELD/STATEMENT/ITEM: SOURCE NOT RECOGNIZED.

The indicated report field specification has an invalid source code. Correct the report program before running it again.

120 <file letter> FILE: ERROR IN FIELD OFFSET AND/OR LENGTH.

The control file key field specification for this file includes a field whose offset and length exceed the bounds of the primary-file record.

121 <file letter> FILE: KEY LENGTH LESS THAN LENGTH OF KEY FIELDS IN PROGRAM.

The specification of the fields in the control file to be used in accessing the primary file define a partial key that is actually longer than the primary-file key. Designated data from the control file should comprise a prefix in the primary-file key or, at the most, the full key (but not more than the key).

122 <file letter> FILE: ERROR IN FIELD OFFSET AND/OR LENGTH.

The offset or length into the indicated file, used in defining a break field, exceeds the bounds of the record of that file. Correct the problem before attempting to run the report again.

123 PROGRAM LIBRARY FILE ERROR.

The program library control record was not found. Make sure that the designated file is in fact a program library file or that the correct name has been used when installing the library.

124 PROGRAM LIBRARY FILE ERROR.

The program library control record was not found. Make sure that the designated file is in fact a program library file or that the correct name has been used on the direct selection form in DMRUN.

130 IDENTIFIER ALREADY USED. ENTER A DIFFERENT IDENTIFIER.

An attempt has been made to write a record in the designated file with the same key as an existing record. Reenter the record with a unique key.

132 B20 SYSTEM ERROR.

A forms run-time software error has occurred while attempting to refresh the form. This is a software error. Try initializing the system. The operating system error code may provide an indication of the nature of the problem.

133 THIS ENTRY MUST BE ALL NUMBERS.

An attempt has been made to read invalid data from a field on the form. Since this does not directly involve the entry of the field, the data cannot be corrected by the operator. The program or form must be corrected.

135 THIS ENTRY MUST BE ALL NUMBERS.

A default value in a form field is defined in the program as a numeric field but contains non-numeric characters. Correct the form or the field description before attempting to run the program again.

136 B20 SYSTEM ERROR.

A video-access software error occurred while attempting to clear the message line (bottom line) on the screen. Try reloading the B20 software or initializing the system. The operating system error code may provide an indication of the nature of the problem.

137 B20 SYSTEM ERROR.

A video-access software error occurred while attempting to retrieve data from the screen. Try initializing the B20 or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

138 B20 SYSTEM ERROR.

A forms run-time software error occurred while attempting to retrieve data from the form. Try initializing the B20 or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

139 <file letter> FILE: KEY LENGTH LESS THAN LENGTH OF KEY FIELDS IN PROGRAM.

The data you have entered contains a key which does not exist in the indicated secondary file. Reenter the record with correct data or cancel the record.

140 ENTRY OUTSIDE NUMERIC RANGE LIMITS. TRY LARGER OR SMALLER.

Your entry is greater than the specified range allows. Reenter an acceptable value.

141 THIS ENTRY IS OUTSIDE THE ALPHABETIC RANGE LIMITS.

Your entry is outside the specified range. Reenter an acceptable value.

142 NUMERIC ENTRY MAY NOT BE BLANK OR ZERO. REENTER.

This field is mandatory. It cannot be left blank or contain zero. Reenter a non-blank and nonzero value.

146 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

The source in the indicated field specification is invalid or unavailable. Correct the program before attempting to run it again.

147 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

The number associated with the source code in the indicated field specification is invalid for that source code. Correct the program before attempting to run it again.

149 NO INITIAL PROGRAM SPECIFIED FOR THIS PROGRAM LIBRARY.

The program library has no initial program specified. Use the program library maintenance function in DMCREATE to specify the name of the program to be initially run when that program library is selected.

150 DELETE NOT ALLOWED IN THIS PROGRAM.

This program does not permit the deletion of records. If records must be deleted then change the program to allow deletion or set up another similar program for record deletion.

151 ENTRIES MAY NOT BE CHANGED WITH THIS PROGRAM.

This program only allows an existing record to be deleted, not changed. Delete the record or cancel to preserve the record.

152 <field number> FIELD: SOURCE REFERENCE PRECEDES ENTRY OF COMPLETE KEY.

The indicated field specification has referenced as a source a file which has not yet been read. Secondary files are only read when their keys have been completely entered. Any specification of a secondary file source can therefore be made only after all of its key fields have been declared.

153 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

The program declared as the record description program for the indicated file does not have a primary file. Correct the program before continuing.

154 ENTRIES NOT FOUND.

The indicated secondary file was not found. Make sure this file is present and in the correct directory on the correct volume before attempting to run the program again.

156 <statement number> STATEMENT: INVALID LINE NUMBER IN GOTO STATEMENT.

The indicated statement in one of the program functions has a GOTO instruction which has an invalid destination parameter. Correct the program before attempting to run it again.

157 <statement number> STATEMENT: GOTO BEYOND END OF FUNCTION.

The indicated statement in one of the program functions has a GOTO instruction which addresses a nonexistent statement as its destination. Correct the program before attempting to run it again.

158 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program functions has a DELETE instruction with a parameter other than a file letter (i.e., other than A-I).

159 <statement number> STATEMENT: OPERATOR NOT RECOGNIZED.

The indicated statement in one of the program functions has an assignment instruction in which the operator code is other than an equal sign (=). The error could also be caused by declaring other than an assignment statement in which the first component was not a valid keyword. Correct the program before attempting to run it again.

160 <statement number> STATEMENT: MISSING VARIABLE.

The indicated statement in one of the program functions has an assignment instruction in which nothing is being assigned. In other words the statement has nothing to the right of the equal sign (=) other than possibly a comment. Correct the program before attempting to run it again.

161 <statement number> STATEMENT: STATEMENT CONTAINS UNRECOGNIZED CHARACTER(S).

The indicated statement in one of the program functions has an assignment instruction containing spurious data. Correct the program before attempting to run it again.

162 <statement number> STATEMENT: MISSING VARIABLE.

The indicated statement in one of the program functions has an incomplete assignment instruction. Correct the program before attempting to run it again.

163 <statement number> STATEMENT: DATA TYPE MISMATCH.

The indicated statement in one of the program functions has an IF instruction in which two data items of different types are being compared. Correct the program before attempting to run it again.

164 <statement number> STATEMENT: MISSING VARIABLE.

The indicated statement in one of the program functions has an incomplete IF instruction. Correct the program before attempting to run it again.

165 <statement number> STATEMENT: MISSING VARIABLE.

The indicated statement in one of the program functions has an incomplete IF instruction. Correct the program before attempting to run it again.

166 <statement number> STATEMENT: MISSING VARIABLE.

The indicated statement in one of the program functions has an incomplete IF instruction. Correct the program before attempting to run it again.

167 <statement number> STATEMENT: MISSING VARIABLE.

The indicated statement in one of the program functions has an incomplete IF instruction. Correct the program before attempting to run it again.

168 <statement number> STATEMENT: CONDITIONAL OPERATOR NOT RECOGNIZED.

The indicated statement in one of the program functions has an IF instruction with an invalid conditional operator. Correct the program before attempting to run it again.

169 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program functions has an IF instruction which addresses a nonexistent statement as its destination. Correct the program before attempting to run it again.

170 DM PROGRAM CORRUPT: PROGRAM UNUSABLE.

A Data Manager error has occurred in which an invalid function type has been encountered. Your software supplier should be notified of the problem.

171 <file letter> FILE: FILE REQUIRED FOR FUNCTION STATEMENT HAS NOT BEEN READ.

Data from the indicated file has been referenced in one of the program's functions but the file has not been read. If secondary files are to be referenced in a function, they should be declared to be critical. If the file was declared to be critical, make sure that all of its key fields have been declared in the primary record and that they match, in length, their counterparts in the secondary-file key.

172 <file letter> FILE: ERROR IN FIELD OFFSET AND/OR LENGTH.

The indicated field in one of the file records has a declared length which is incorrect and exceeds the record bounds. Locate the problem and fix it before attempting to run the program again.

173 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced an X Register which is not accessible. Correct the program before attempting to run it again.

174 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced an invalid source code. Correct the program before attempting to run it again.

175 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced an invalid source designation. Correct the program before attempting to run it again.

176 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced a source with an incorrect field number.

177 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced a nonexistent program register. Correct the program before attempting to run it again.

178 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced a nonexistent report register. Correct the program before attempting to run it again.

179 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced a nonexistent X Register. Correct the program before attempting to run it again.

180 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced a nonexistent total register. Correct the program before attempting to run it again.

181 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced an invalid source code. Correct the program before attempting to run it again.

183 <file letter> FILE: ERROR IN FIELD OFFSET AND/OR LENGTH.

The indicated field in one of the file records has a declared length which is incorrect and exceeds the record bounds. Locate the problem and fix it before attempting to run the program again.

185 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced a nonexistent X Register. Correct the program before attempting to run it again.

186 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced an invalid source code. Correct the program before attempting to run it again.

187 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced an alpha literal which was not terminated with a quote. Correct the program before attempting to run it again.

188 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced an alpha literal which is within matching quotation marks. Correct the program before attempting to run it again.

189 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced an invalid source code. Correct the program before attempting to run it again.

190 <statement number> STATEMENT: DATA TYPE MISMATCH.

The indicated statement in one of the program's functions has attempted to make an assignment involving data of unlike types. Correct the program before attempting to run it again.

191 <statement number> STATEMENT: OPERATOR NOT RECOGNIZED.

The indicated statement in one of the program's functions is an assignment with an invalid arithmetic operator. Correct the program before attempting to run it again.

192 <statement number> STATEMENT: DATA TYPE MISMATCH.

The indicated statement in one of the program's functions has attempted to perform arithmetic between data of unlike types. Correct the program before attempting to run it again.

193 STANDARD DESCRIPTION OUT OF RANGE.

The program contains a standard description which is 0 or greater than 28. Correct the program before attempting to run it again.

194 <number> FIELD/STATEMENT/ITEM: ARITHMETIC OVERFLOW.

The indicated field has received data that was too large for it and would result in the truncation of significant digits. The field should be made larger or such large data should be prevented from occurring.

200 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

201 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

202 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

203 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

204 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

205 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

206 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

207 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

208 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

209 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

210 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

211 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

212 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

213 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

214 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

215 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

216 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

217 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

218 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

219 THIS ENTRY MUST BE ALL NUMBERS.

Non-numeric data has been entered in a numeric field. Reenter using numeric digits only or a sign or a decimal where allowed.

220 ENTRY MAY NOT CONTAIN A "-" OR A "+". REENTER WITHOUT SIGN.

A minus sign has been entered in an unsigned field. Reenter without the sign or change the field in the program to a signed field.

221 TOO MANY WHOLE NUMBERS ENTERED BEFORE THE DECIMAL POINT.

The whole number portion of the data entered is too large for the field as specified. Reenter the correct value or change the field specifications in the program to allow for larger data to be entered.

222 TOO MANY NUMBERS ENTERED AFTER THE DECIMAL POINT.

Too many digits have been entered beyond the decimal point. Reenter the correct value or change the field specifications in the program to allow for larger data to be entered.

223 <field number> FIELD: FIELD NAME IN PROGRAM NOT FOUND ON FORM.

A forms error has occurred. The most likely cause is a reference to a field name that was not declared on the form. Make sure the name in the indicated field corresponds to a field on the form. Check the operating system error code.

225 <field number> FIELD: FIELD NAME IN PROGRAM NOT FOUND ON FORM.

A forms error has occurred. The most likely cause is a reference to a field name that was not declared on the form. Make sure the name in the indicated field corresponds to a field on the form. Check the operating system error code.

226 DM PROGRAM CORRUPT: PROGRAM UNUSABLE.

A Data Manager error has occurred while displaying a standard menu form. Your software supplier should be notified.

227 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

228 YOUR SELECTION IS NOT AVAILABLE. PLEASE ENTER ANOTHER.

An invalid selection has been made. Reenter a valid number corresponding to the available selections on the screen.

229 THIS ENTRY MUST BE ALL NUMBERS.

A non-numeric entry has been made for the selection. This data must be all numbers and must correspond to the available selections on the screen.

230 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

231 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software. Try initializing the B20 or reloading DMRUN. The operating system error code may provide an indication of the nature of the problem.

232 LIBRARY ID/PROGRAM NAME IS REQUIRED.

The library ID is a required entry for program type 1 (Data Manager program) or 7 (ISAM Reorganize).

235 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has referenced a file letter other than A-I. Correct the program before attempting to run it again.

236 <statement number> STATEMENT: ARITHMETIC OPERATOR NOT RECOGNIZED.

The indicated statement in one of the program's functions has used an invalid arithmetic operator. Correct the program before attempting to run it again.

237 B20 SYSTEM ERROR.

A memory management system error has occurred. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

238 <number> FIELD/STATEMENT/ITEM: SOURCE NOT RECOGNIZED.

The indicated field specification contains a reference to a secondary file for which no record description program name was specified. Correct the program before running it again.

239 B20 SYSTEM ERROR.

A timer management system error has occurred. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

240 B20 SYSTEM ERROR.

A timer management system error has occurred. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

241 B20 SYSTEM ERROR.

An invalid system date has been retrieved. The B20 should be initialized before continuing.

242 B20 SYSTEM ERROR.

An invalid system date has been retrieved. The B20 should be initialized before continuing.

243 B20 SYSTEM ERROR.

An invalid system date has been retrieved. The B20 should be initialized before continuing.

245 B20 SYSTEM ERROR.

A timer management system error has occurred. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

246 B20 SYSTEM ERROR.

A timer management system error has occurred. Try initializing the B20 or replacing system software. The operating system error code may provide an indication of the nature of the problem.

247 B20 SYSTEM ERROR.

An invalid system time has been retrieved. The B20 should be initialized before continuing.

248 B20 SYSTEM ERROR.

An invalid system time has been retrieved. The B20 should be initialized before continuing.

249 B20 SYSTEM ERROR.

An invalid system time has been retrieved. The B20 should be initialized before continuing. operating system error code may provide an indication of the nature of the problem.

250 PLEASE REENTER DATE. CHECK FOR PROPER M, D, Y SEQUENCE.

The date you have entered is invalid. Reenter a valid date.

251 PLEASE REENTER DATE. CHECK FOR PROPER M, D, Y SEQUENCE.

The date you have entered is invalid. Reenter a valid date.

254 DM PROGRAM CORRUPT: PROGRAM UNUSABLE.

The program has an invalid type code. This is a Data Manager problem and should be reported to your software supplier.

259 LINE PRINTER NOT READY.

The line printer is not ready.

260 LINE PRINTER ERROR.

A system error occurred while displaying a report line on the screen. Try initializing the B20 before trying again. The operating system error code may provide an indication of the nature of the error.

264 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has an ERROR instruction which contains an error number other than 1-5. Correct the program before attempting to run it again.

265 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement in one of the program's functions has an ERROR instruction which contains an error number other than 1-5. Correct the program before attempting to run it again.

267 DM ENTRY FORM ERROR.

The library selection form is corrupt. Try reloading the system forms file, DMRFORMS. Otherwise, try reloading DMRUN. The operating system error code may provide an indication of the nature of the problem.

268 OUTPUT EDIT DESCRIPTION DOESN'T MATCH DATA TYPE.

In the indicated report specification, the output description is incompatible with the input field type. Correct the program before attempting to run it again.

272 OUTPUT EDIT DESCRIPTION DOESN'T MATCH DATA TYPE.

In the indicated report specification, the output description is incompatible with the input field type. Correct the program before attempting to run it again.

273 OUTPUT EDIT DESCRIPTION DOESN'T MATCH DATA TYPE.

In the indicated report specification, the output description is incompatible with the input field type. Correct the program before attempting to run it again.

274 OUTPUT EDIT DESCRIPTION DOESN'T MATCH DATA TYPE.

In the indicated report specification, the output description is incompatible with the input field type. Correct the program before attempting to run it again.

275 OUTPUT EDIT DESCRIPTION DOESN'T MATCH DATA TYPE.

In the indicated report specification, the output description is incompatible with the input field type. Correct the program before attempting to run it again.

276 LIBRARY VERSION INCOMPATIBLE.

This version of the program library is incompatible with the program accessing it.

277 PROGRAM VERSION INCOMPATIBLE.

This version of the program is incompatible with the program accessing it. The program will have to be reentered using the current release of Data Manager.

278 DMRDATA VERSION INCOMPATIBLE.

This version of DMRDATA is incompatible with this version of DMRUN. Make sure that the current release of Data Manager is not mixed with previous releases.

279 FORM HAS BEEN REQUESTED AND NO FORM FILE HAS BEEN DECLARED.

A forms editor form has been requested, but no forms library was declared in the control record of the program library. Use library maintenance in DMCREATE to declare the name of the associated forms library.

280 ENTRY VALIDATION ERROR <message number>.

This is an application-generated error. It results from executing an ERROR instruction in a program function. The number associated with the ERROR instruction determines which of the five error texts is displayed.

281 LIBRARY VERSION INCOMPATIBLE.

This version of the program library is incompatible with the program accessing it. The program library will have to be reentered using the current release of Data Manager.

282 INVALID SYSTEM DATE CODE IN SYSTEM FILE CONTROL INFORMATION.

The date type in DMRDATA is invalid.

283 <statement number> STATEMENT: INVALID DATE ENCOUNTERED DURING PROCESSING.

Either the date type in DMRDATA is invalid, or the date being converted to Julian form by the indicated statement is invalid.

284 <statement number> STATEMENT: UNUSABLE AGE PERIOD LENGTH IN X6.

The aging period loaded into X6 is invalid. The program must be corrected to load a valid aging period into X6 before attempting to run it again.

285 <statement number> STATEMENT: INVALID DATE ENCOUNTERED DURING PROCESSING.

An invalid date has been used in the AGE instruction. Either the system date or the date in X4 is invalid. Check your program to make sure that these dates are valid.

286 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while data was entered. Try replacing the DMRFORMS file. Otherwise, try initializing the B20, reloading the system software, or reloading DMRUN. The operating system error code may provide an indication of the nature of the problem.

287 DMRDATA FILE ERROR.

A write error has occurred while writing to DMRDATA. Try using a backup copy of DMRDATA. The problem may be caused by a disk hardware problem. The operating system error code may provide an indication of the nature of the problem.

288 <statement number> STATEMENT: UNUSABLE AGE PERIOD LENGTH IN X6.

The aging period loaded into register X6 is invalid. You must correct the program by loading a valid aging period into X6 before attempting to run it again. The aging period cannot be zero.

289 DMRDATA FILE ERROR.

A write error has occurred while writing to DMRDATA. Try using a backup copy of DMRDATA. The problem may be caused by a disk hardware problem. The operating system error code may provide an indication of the nature of the problem.

290 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying data. Try initializing the B20 or reloading the system software. Otherwise, try replacing the DMRFORMS file with a backup copy. The operating system error code may provide an indication of the nature of the problem.

291 DM ENTRY FORM ERROR.

A system error has occurred in the video-access software. Try initializing the B20 or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

292 DM ENTRY FORM ERROR.

A Data Manager system form could not be found. Try replacing the DMFORMS file. The operating system error code may provide an indication of the nature of the problem.

293 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

294 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while data was entered. Try replacing the DMRFORMS file. Otherwise, try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

295 DM ENTRY FORM ERROR.

Invalid data has been entered. Reenter valid data.

296 DM ENTRY FORM ERROR.

Invalid data has been entered. Reenter valid data.

297 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

298 LIBRARY SELECTION MENU ALREADY FULL.

Ten applications have already been installed.

299 THIS ENTRY MUST BE ALL NUMBERS.

Non-numeric data has been entered. Reenter the field, using only numeric digits.

300 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

301 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

302 DMRDATA FILE ERROR.

A read error has occurred while reading DMRDATA. Try replacing DMRDATA. It could also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

303 DMRDATA FILE ERROR.

A write error has occurred while writing to DMRDATA. Try replacing DMRDATA. It could also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

304 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

305 B20 SYSTEM ERROR.

A system error has occurred in the forms run-time software while data was entered. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

306 B20 SYSTEM ERROR.

A system error has occurred in the forms run-time software while displaying data. Try initializing the B20, reloading DMRUN, or reloading the system software. Otherwise, try replacing the DMRFORMS file with a backup copy. The operating system error code may provide an indication of the nature of the problem.

307 DMRDATA FILE ERROR.

A write error has occurred while writing DMRDATA. Try replacing DMRDATA. It could also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

308 THIS PROGRAM LIBRARY IS NOT INSTALLED.

The application number you have selected is not installed. Select a valid number.

309 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

310 DM ENTRY FORM ERROR.

A Data Manager system form could not be found. Try replacing the DMRFORMS file. The operating system error code may provide an indication of the nature of the problem.

311 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

312 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while data was entered. Try replacing the DMRFORMS file. Otherwise, try initializing the B20, reloading DMRUN,or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

313 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

314 THIS PROGRAM LIBRARY IS NOT INSTALLED.

The application number you have selected is not installed. Select a valid number.

315 DMRDATA FILE ERROR.

A read error has occurred while reading DMRDATA. Try replacing DMRDATA. It could also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

316 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying data. Try initializing the B20, reloading DMRUN, or reloading the system software. Otherwise, try replacing the DMRFORMS file with a backup copy. The operating system error code may provide an indication of the nature of the problem.

317 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying data. Try initializing the B20, reloading DMRUN, or reloading the system software. Otherwise, try replacing the DMRFORMS file with a backup copy. The operating system error code may provide an indication of the nature of the problem.

318 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may also provide an indication of the nature of the problem.

319 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while data was entered. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

320 THIS PROGRAM LIBRARY IS NOT INSTALLED.

The application number you have selected is not installed. Select a valid number.

321 DMRDATA FILE ERROR.

A write error has occurred while writing to DMRDATA. Try replacing DMRDATA. It could also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

322 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying data. Try initializing the B20, reloading DMRUN, or reloading the system software. Otherwise try replacing the DMRFORMS file with a backup copy. The operating system error code may provide an indication of the nature of the problem.

323 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

324 DM ENTRY FORM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

325 DM ENTRY FORM ERROR.

A Data Manager system form could not be found. Try replacing the DMRFORMS file. Otherwise, try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the problem.

326 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

327 DM ENTRY FORM ERROR.

A system error occurred in the forms run-time software while data was entered. Try replacing the DMRFORMS file. Otherwise, try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

328 INVALID DATA ENTERED.

Invalid data has been entered. Reenter valid data.

329 THIS PROGRAM LIBRARY IS NOT INSTALLED.

The application number you have selected is not installed. Select a valid number.

330 INVALID DATA ENTERED.

Invalid data has been entered. Reenter valid data.

331 THIS PROGRAM LIBRARY IS NOT INSTALLED.

The application number you have selected is not installed. Select a valid number.

332 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

333 THIS PROGRAM LIBRARY IS NOT INSTALLED.

The application number you have selected is not installed. Select a valid number.

334 DMRDATA FILE ERROR.

A system error has occurred while closing DMRDATA. The operating system may provide an indication of the nature of the problem.

335 TERMINATE DMRUN ON OTHER STATIONS TO DO LIBRARY MAINTENANCE.

DMRDATA is in use and cannot be accessed for modification at this time. When performing library maintenance, no other station in the cluster may by running DMRUN.

336 DMRDATA FILE ERROR.

A read error has occurred while reading DMRDATA. Try replacing DMRDATA. It could also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

337 DM ENTRY FORM ERROR.

A Data Manager system form could not be found. Try replacing the DMRFORMS file. The operating system error code may provide an indication of the nature of the problem.

338 DM ENTRY FORM ERROR.

A Data Manager system form could not be found. Try replacing the DMRFORMS file. The operating system error code may provide an indication of the nature of the problem.

339 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

340 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

341 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

342 DMRDATA FILE ERROR.

A read error has occurred while reading DMRDATA. Try replacing DMRDATA. It could also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

343 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying data. Try initializing the B20, reloading DMRUN, or reloading the system software. Otherwise, try replacing the DMRFORMS file with a backup copy. The operating system error code may provide an indication of the nature of the problem.

344 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while entering data. Try replacing the DMFORMS file. Otherwise, try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

345 THIS ENTRY MUST BE ALL NUMBERS.

Non-numeric data has been entered. Reenter the field, using numeric digits only.

346 YOUR SELECTION IS NOT AVAILABLE. PLEASE ENTER ANOTHER.

The selection you have made is not available. Reenter a valid number corresponding to one of the selections on the screen.

347 DMRDATA FILE ERROR.

A system error has occurred while closing DMRDATA.

348 DMRDATA FILE ERROR.

A system error has occurred while reopening DMRDATA for input only.

349 DM ENTRY FORM ERROR.

A Data Manager system form could not be found. Try replacing the DMRFORMS file. The operating system error code may provide an indication of the nature of the problem.

350 B20 SYSTEM ERROR.

A system error has occurred in the video-access software. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

351 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. Th operating system error code may provide an indication of the nature of the problem.

352 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while entering data. Try replacing the DMFORMS file. Otherwise, try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

353 THIS ENTRY MUST BE ALL NUMBERS.

Non-numeric data has been entered. Reenter the field, using numeric digits only.

354 YOUR SELECTION IS NOT AVAILABLE. PLEASE ENTER ANOTHER.

The selection you made is not available. Reenter a valid number corresponding to one of the selections on the screen.

355 YOUR SELECTION IS NOT AVAILABLE. PLEASE ENTER ANOTHER.

The selection you have made is not available. Reenter a valid number corresponding to one of the selections on the screen.

356 FILE NAME AND DESCRIPTION REQUIRED FOR INSTALLATION.

When installing an application, both the description of the application and its file ID are required. Make sure both are entered.

357 <report item number> ITEM: REPORT TEXT NOT IN TEXT TABLE.

The indicated report specification contain a reference to a nonexistent report text. Correct the program before continuing.

359 CANNOT DELETE PRIMARY FILE.

ISAM error while opening the primary file. The operating system error code may provide an indication of the nature of the problem.

360 CANNOT DELETE PRIMARY FILE.

A Data Manager error has occurred, preventing the deleting of the primary file. The operating system error code may provide an indication of the nature of the problem.

361 CANNOT START TRANSACTION.

An ISAM error has occurred while starting a transaction. Try initializing the B20. The operating system error code may provide an indication of the nature of the problem.

362 CANNOT END TRANSACTION.

An ISAM error has occurred while ending a transaction. Try initializing the B20. The operating system error code may provide an indication of the nature of the problem.

363 CANNOT SAVE ZIP RETURN INFORMATION.

A system write error has occurred while writing the chain return file. This could be caused by a disk hardware error. Try initializing the B20 or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

364 CANNOT SAVE ZIP RETURN INFORMATION.

A system error occurred while opening the chain return file. This could be caused by a disk hardware error. Try initializing the B20 or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

366 BAD ZIP RETURN FILE; CANNOT ACCESS.

The chain return file is corrupt and cannot be interpreted. Automatic restart will not occur.

367 CANNOT DELETE ZIP RETURN FILE.

A system error occurred while deleting the chain return file. This file must be deleted manually. This file will have the suffix DMR.TMP.

368 BAD ZIP RETURN FILE; CANNOT ACCESS.

A read error occurred while reading the chain return file. Automatic restart will not occur.

369 BAD ZIP RETURN FILE; CANNOT ACCESS.

A system error occurred while opening the chain return file. Automatic restart will not occur.

370 DM ENTRY FORM ERROR.

A Data Manager form could not be found. Try replacing the system forms file, DMFORMS. The operating system error code may provide an indication of the nature of the problem.

371 INCORRECT PROGRAM TYPE FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program is not a Maintenance, Real-time Update with a transaction log, or a Menu program and therefore has no primary file. Correct the program before attempting to run it again.

372 INCOMPATIBLE VERSION FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program is not is not the same version as DMRUN. The record description program must be re-created or a different program referenced.

373 INCORRECT PROGRAM TYPE FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program was not a Maintenance, Real-time Update with a transaction log, or a Menu program and therefore has no primary file. Correct the program before attempting to run it again.

374 INCOMPATIBLE VERSION FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program is not the same version as DMRUN. The record description program must be re-created or a different program referenced.

375 INCORRECT PROGRAM TYPE FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program was not a Maintenance, Real-time Update with a transaction log, or a Menu program and therefore has no primary file. Correct the program before attempting to run it again.

376 INCOMPATIBLE VERSION FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program is not the same version as DMRUN. The record description must be re-created or a different program referenced.

377 INCORRECT PROGRAM TYPE FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program was not a Maintenance, Real-time Update with a transaction log, or Menu program and therefore has no primary file. Correct the program before attempting to run it again.

378 INCOMPATIBLE VERSION FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program is not of the same version as DMRUN. The record description must be re-created or a different program referenced.

379 INCORRECT PROGRAM TYPE FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program was not a Maintenance, Real-time Update with a transaction log or a Menu program and therefore has no primary file. Correct the program before attempting to run it again.

380 INCOMPATIBLE VERSION FOR RECORD DESCRIPTION PROGRAM.

The program specified as a record description program is not of the same version as DMRUN. The record description must be re-created or a different program referenced.

381 DMRDATA FILE ERROR.

A read error has occured while reading DMRDATA. Try replacing the DMRDATA file. The problem may also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

382 CANNOT ACCESS FILES FROM REPORT HEADER OR TRAILER.

The indicated report specification in a report header or trailer contains a reference to a file field. Files cannot be accessed in the report header or trailer. Correct the program before attempting to run it again.

385 OUTPUT DEVICE CODE MUST BE 1-5.

The output device code for reports must be the number 1, 2, 3, 4, or 5. Correct the program.

386 CANNOT CLOSE LINE PRINTER.

A system error has occurred while closing the printer file. If the job was being spooled, make sure there was enough disk space available. The operating system error code may provide an indication of the nature of the problem.

389 CANNOT PRINT DIRECT. (PRESS GO TO SPOOL, FINISH TO END.)

A request for direct printing of a report has been specified on a workstation that cannot support a printer.

390 CANNOT PRINT DIRECT. (PRESS GO TO SPOOL, FINISH TO END.)

A request for direct printing of a report has been specified on a workstation that cannot support a printer.

391 CANNOT DISCONNECT PRINTER FROM SPOOLER.

The spooler has indicated that it cannot release the printer for direct printing. The operating system error code may provide an indication of the nature of the problem.

392 CANNOT DISCONNECT PRINTER FROM SPOOLER.

The spooler has indicated that it cannot release the printer for direct printing. The operating system error code may provide an indication of the nature of the problem.

394 CANNOT OPEN LINE PRINTER.

A system error has occurred while opening the printer file. Make sure the printer configuration file is present. The operating system error code may provide an indication of the nature of the error.

395 CANNOT RECONNECT PRINTER TO SPOOLER.

A system error has occurred while attempting to reconnect the printer to the spooler. The operating system error code may provide an indication of the nature of the problem.

396 PROGRAM TO BE INITIATED NOT ON DISK.

An attempt was made to chain a file which could not be accessed. The operating system error code may provide an indication of the nature of the problem.

397 CANNOT INITIATE THIS NON-DATA MANAGER PROGRAM.

A system error occurred while attempting to chain a program. The operating system error code may provide an indication of the nature of the problem.

398 SECONDARY KEY FIELD NOT INCLUDED IN RECORD.

The indicated secondary-file key field specification references a field which is not included in the record. This is not allowed in a Batch Update, Report, or Inquiry program, nor if the file is to be referenced in a Repeating function. Correct the program before attempting to run it again.

399 ENTRIES REQUIRED BUT NOT FOUND.

A critical secondary-file record with this key cannot be found. Reenter the record or cancel it.

400 <statement number> STATEMENT: INVALID DATE ENCOUNTERED DURING PROCESSING.

An attempt was made to convert an invalid date to century Julian format.

401 <statement number> STATEMENT: INVALID DATE ENCOUNTERED DURING PROCESSING.

An attempt was made to convert an invalid date to century Julian format.

402 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

403 CANNOT REORGANIZE FILE - CHECK IF VALID FILE' EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

404 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

405 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

406 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

407 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

408 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

409 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

410 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

411 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

412 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

413 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

414 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

415 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

416 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while setting up parameters for reorganizing an ISAM file. The operating system error code may provide an indication of the nature of the problem.

418 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while attempting to chain ISAMReorganize.Run. The operating system error code may provide an indication of the nature of the problem.

419 SUBMIT FILE NOT ON DISK.

An attempt was made to submit a file which cannot be accessed. The operating system error code may provide an indication of the nature of the problem.

420 CANNOT INITIATE SUBMIT.

A system error occurred while setting up parameters for submitting a file. The operating system error code may provide an indication of the nature of the problem.

421 CANNOT INITIATE SUBMIT.

A system error occurred while setting up parameters for submitting a file. The operating system error code may provide an indication of the nature of the problem.

422 CANNOT INITIATE SUBMIT.

A system error occurred while attempting to chain Submit.Run. The operating system error code may provide an indication of the nature of the problem.

423 COBOL PROGRAM NOT PRESENT ON DISK.

An attempt was made to chain a file which could not be accessed. The operating system error code may provide an indication of the nature of the problem.

424 CANNOT RUN COBOL PROGRAM - CHECK COBOL RUN TIME.

A system error occurred while setting up parameters for running a COBOL program. The operating system error code may provide an indication of the nature of the problem.

425 CANNOT RUN COBOL PROGRAM - CHECK COBOL RUN TIME.

A system error occurred while setting up parameters for running a COBOL program. The operating system error code may provide an indication of the nature of the problem.

426 CANNOT RUN COBOL PROGRAM - CHECK COBOL RUN TIME.

A system error occurred while attempting to chain a program. The operating system error code may provide an indication of the nature of the problem.

427 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

428 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while entering data. Try replacing the DMRFORMS file. Otherwise try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

429 THIS ENTRY MUST BE ALL NUMBERS.

Non-numeric data has been entered. Reenter the field using only numeric digits.

430 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

Error while attempting to reorganize an ISAM file. Make sure that the run file <SYS>ISAMReorganize.RUN is present.

431 DM ENTRY FORM ERROR.

A system error has occurred in the forms run-time software while displaying a form. Try initializing the B20, reloading DMRUN, or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

433 DMRDATA FILE ERROR.

A read error has occurred while reading DMRDATA. Try replacing DMRDATA. It could also be a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

434 THIS PROGRAM LIBRARY IS NOT INSTALLED.

The application number you have selected is not installed. Select a valid number.

435 CANNOT REORGANIZE FILE - CHECK IF VALID FILE EXISTS.

A system error occurred while attempting to reorganize an ISAM file. The operating system error code may provide an indication of the nature of the problem.

436 OUTPUT DEVICE CODE MUST BE 1-5.

The output device type requested by the program is not 1-5. Correct the program before attempting to run it again.

437 CANNOT SORT FILE SPECIFIED - CHECK FOR VALID FILE.

A system error occurred while preparing to sort the control file. The operating system error code may provide an indication of the nature of the problem.

438 CANNOT SORT FILE SPECIFIED - CHECK FOR VALID FILE.

A system error occurred while sorting the control file. The operating system error code may provide an indication of the nature of the problem.

439 CANNOT SORT FILE SPECIFIED - CHECK FOR VALID FILE.

A system error occurred while sorting the control file. The operating system error code may provide an indication of the nature of the problem.

440 PROGRAM TYPE IN LINK SPECIFICATIONS MUST BE 1-8.

The previous program has attempted to link to a program with an invalid program type code. The program type must be a number from 1 to 8. Correct that program's linkage before attempting to use it again.

441 ISAM NOT INSTALLED. INSTALL ISAM BEFORE CONTINUING.

Isam is not installed. Isam is required software for Data Manager. Install ISAM before attempting to run DMRUN again.

442 DM PROGRAM NOT FOUND OR INCOMPLETE.

An attempt was made to load an incomplete Data Manager program. Complete the program before attempting to run it again.

443 CANNOT PRINT DIRECT FROM THIS STATION.

A request for direct printing of a report has been specified on a workstation that cannot support a printer.

444 <report item number> ITEM: SOURCE FIELD NOT INCLUDED IN RECORD.

The indicated report specification has referenced a field which was not included in the record. Correct the program before attempting to run it again.

445 <report item number> ITEM: SOURCE FIELD NOT INCLUDED IN RECORD.

The indicated report specification has referenced a field which was not included in the record. Correct the program before attempting to run it again.

446 <report item number> ITEM: SOURCE FIELD NOT INCLUDED IN RECORD.

The indicated report specification has referenced a field which was not included in the record. Correct the program before attempting to run it again.

448 CANNOT SORT FILE SPECIFIED - CHECK FOR VALID FILE.

A read error occurred while attempting to read a sorted record. This may mean a disk hardware error. The operating system error code may provide an indication of the nature of the problem.

449 PRIMARY FILE KEY FIELD NOT INCLUDED IN CONTROL RECORD.

A Key Field specification for access to the primary file has referenced a control file field that is not included in the record. Correct the program before continuing.

450 <report item number> ITEM: SOURCE FIELD NOT INCLUDED IN RECORD.

A Data Manager error has resulted in an attempt to store data in a record when that data was not included in record. This problem should be reported to your software supplier.

451 CANNOT SPOOL OR PRINT DIRECT - PRESS THE FINISH KEY TO EXIT.

An attempt has been made to print a report via an invalid spooler queue.

453 CANNOT SPOOL OR PRINT DIRECT - PRESS THE FINISH KEY TO EXIT.

An attempt has been made to print a report via an invalid spooler queue.

454 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

The indicated field specification references a nonexistent secondary file. The program should be corrected before it is run again.

455 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

The indicated field specification references a nonexistent secondary file. The program should be corrected before it is run again.

456 <number> FIELD/STATEMENT/ITEM: SOURCE NOT RECOGNIZED.

The indicated report specification references a nonexistent program register. The program should be corrected before an attempt is made to run it again.

457 INVALID CONTROL BREAK FIELD.

A control break field specification has referenced a nonexistent program register. The program should be corrected before an attempt is made to run it again.

458 INVALID CONTROL BREAK FIELD.

An invalid control break field has been specified. The only valid control break specifications are program registers and field identifiers. The program should be corrected before an attempt is made to run it again.

459 <statement number> STATEMENT: REFERENCE TO FIELD THAT IS NOT INCLUDED IN RECORD.

This statement has referenced a file field that is not included in the record. Such fields do not exist in the record and may not be referenced. Correct the program before continuing.

460 <statement number> STATEMENT: REFERENCE TO FIELD THAT IS NOT INCLUDED IN RECORD.

This statement has referenced a file that is not included in the record. Such fields do not exist in the record and cannot be referenced. Correct the program before continuing.

461 <statement number> STATEMENT: REFERENCE TO UNDECLARED SECONDARY FILE.

The indicated function statement has referenced an undeclared secondary file. The program should be corrected before another attempt is made to run it.

462 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can be successfully run.

463 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can be successfully run.

464 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can be successfully run.

465 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can be successfully run.

468 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can be successfully run.

470 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can be successfully run.

471 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can be successfully run.

472 PRINT LINE OVERFLOW.

The indicated report specification exceeds the maximum print line length of 250 characters. The program must be corrected before it can be successfully run.

474 <field number> FIELD: ALPHA FIELD CANNOT BE INCREMENTED.

The indicated field specification describes an alpha field which has an automatic increment associated with it. The automatic increment capability can only be used on numeric fields. The program must be corrected before it can be successfully run.

477 <file letter> FILE: ERROR IN FIELD OFFSET AND/OR LENGTH.

The indicated secondary-file key specification has an offset or length which crosses a field boundary. An offset and length must describe a field or part of a field. It may not describe parts of more than one field. The program must be corrected before another attempt is made to run it.

478 MORE THAN 20 FIELDS DECLARED FOR A STANDARD FORM PROGRAM.

A program having more than twenty fields cannot use a standard form. A Forms Editor form will have to be used.

479 <field number> FIELD: SKIP IF BLANK TO UNDECLARED FIELD.

The indicated field specification contains a "skip if blank" with a destination that addresses a nonexistent field specification. The program must be corrected before it can be successfully used.

480 NO PRIMARY FILE HAS BEEN DECLARED.

This report program has a detail element or procedure but no primary file. If there is no primary file the report may only consist of a report header and/or trailer.

482 CONTROL FILE REFERENCES AN UNDECLARED FILE.

This report program has specified a sort but has no control file to sort.

484 <field number> FIELD: CRITICAL FILE NOT READ - KEY SPECIFICATION ERROR.

The indicated field has a source in a critical file that has not been read. This is probably because the key field specifications for that file are incorrect so that the file was not read. The program should be corrected if expected results are to be obtained.

485 <file letter> FILE: CRITICAL FILE NOT READ -- KEY SPECIFICATION ERROR.

The indicated critical file has not been read. This is probably because the key field specifications for that file are incorrect so that the file was not read. The program should be corrected before attempting to run it again.

486 CANCEL CURRENT INPUT BEFORE ATTEMPTING TO FINISH.

The FINISH key may only be used on the first keyboard entered field and if no editing error is waiting to be corrected. To be able to depress FINISH, the input should first be cancelled.

487 <field number> FIELD: A FIELD WITH NO NAME HAS AN AUTOMATIC INCREMENT.

The indicated field specification has neither a name nor an automatic increment. This is illegal. Automatic incrementing is only available on named fields. Correct the program before attempting to run it again.

488 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

The value that your program has caused or allowed to be loaded into X3 as a link index is greater than the size of the link table. If the number is to be entered by the operator, a range check should be used to make sure that only a valid entry is made.

489 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

The value that your program has caused or allowed to be loaded into X3 as a link index is greater than the size of the link table. If the number is to be entered by the operator, a range check should be used to make sure that only a valid entry is made.

490 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

The value that your program has caused or allowed to be loaded into X3 as a link index is greater than the size of the link table. If the number is to be entered by the operator, a range check should be used to make sure that only a valid entry is made.

491 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

The value that your program has caused or allowed to be loaded into X3 as a link index is greater than the size of the link table. If the number is to be entered by the operator, a range check should be used to make sure that only a valid entry is made.

492 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

The value that your program has caused or allowed to be loaded into X3 as a link index is greater than the size of the link table. If the number is to be entered by the operator, a range check should be used to make sure that only a valid entry is made.

493 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

The value that your program has caused or allowed to be loaded into X3 as a link index is greater than the size of the link table. If the number is to be entered by the operator, a range check should be used to make sure that only a valid entry is made.

494 NO ENTRY IN LINK TABLE CORRESPONDS TO VALUE IN X3.

The value that your program has caused or allowed to be loaded into X3 as a link index is greater than the size of the link table. If the number is to be entered by the operator, a range check should be used to make sure that only a valid entry is made.

495 CANNOT OPEN LINE PRINTER.

A system error has occurred while opening the printer file. The operating system error code may provide an indication of the nature of the problem.

496 CANNOT CLOSE LINE PRINTER.

A system error has occurred while closing the printer. The operating system error code may provide an indication of the nature of the error.

497 B20 SYSTEM ERROR.

A system error has occurred while printing a line. The operating system error code may provide an indication of the nature of the error.

498 LINE PRINTER ERROR.

A system error has occurred while printing a line. The operating system error code may provide an indication of the nature of the error.

499 REQ'D ENTRIES IN USE BY ANOTHER STATION. PRESS GO TO RETRY.

A batch update or report program is waiting for a record locked by a transaction at another workstation. When the transaction at the other workstation is completed, press the GO key to restart the batch update or report.

500 REQ'D ENTRIES IN USE BY ANOTHER STATION. PRESS GO TO RETRY.

A batch update or report program is waiting for a record locked by a transaction at another workstation. When the transaction at the other workstation is completed, press the GO key to restart the batch update or report.

501 <field number> FIELD: SOURCE TYPE INCOMPATIBLE WITH DESCRIPTION.

Numeric description assigned to an alpha program register.

502 <field number> FIELD: SOURCE TYPE INCOMPATIBLE WITH DESCRIPTION.

Alpha description assigned to a numeric program register.

503 <field number> FIELD: SOURCE TYPE INCOMPATIBLE WITH DESCRIPTION.

A non-date description assigned to an X Register containing a date.

504 <field number> FIELD: SOURCE TYPE INCOMPATIBLE WITH DESCRIPTION.

Alpha description assigned to a numeric X Register.

505 <field number> FIELD: SOURCE TYPE INCOMPATIBLE WITH DESCRIPTION.

Numeric description assigned to an alpha X Register.

506 INCORRECT PROGRAM TYPE FOR RECORD DESCRIPTION PROGRAM.

The program given as the record description program is not of the correct type. Such a program must be either a Maintenance program, Real-time Update with transaction log or a Menu with a primary file.

507 RECORD DESCRIPTION PROGRAM HAS NO PRIMARY FILE.

The program given as the record description program does not have a primary file.

508 CANCEL CURRENT INPUT BEFORE ATTEMPTING TO FINISH.

The FINISH key cannot be used while the current record is incomplete. Cancel it before attempting to press FINISH.

509 <field number> FIELD: FATAL EDITING ERROR.

An editing error has been detected while processing non-keyboard fields that precede the first keyboard field in the record. There is nothing that the operator can do. The program must be corrected.

510 <field number> FIELD: FATAL EDITING ERROR.

An editing error has been detected while processing non-keyboard fields that precede the first keyboard field in the record. There is nothing that the operator can do. The program must be corrected.

511 <field number> FIELD: FATAL EDITING ERROR.

An editing error has been detected while processing non-keyboard fields that precede the first keyboard field in the record. There is nothing that the operator can do. The program must be corrected.

512 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

Reference has been made to undeclared text of a report. The program must be corrected before another attempt is made to run it.

513 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

Reference has been made to an invalid source. The source must be in the range A-I, P, R, S, T, or X. The program must be corrected before another attempt is made to run it.

514 CANNOT ACCUMULATE NON-NUMERIC FIELD.

An attempt has been made to add a non-numeric field to a report total. Arithmetic can only be performed on numeric data. The program should be corrected before another attempt is made to run it.

515 <statement number> STATEMENT: STATEMENT CONTAINS UNRECOGNIZED CHARACTER(S).

The indicated statement contains an invalid literal. If it is numeric, it can only consist of the digits 0-9 and a plus sign (+) or minus sign (-). The program should be corrected before another attempt is made to run it.

517 <file letter> FILE: DUPLICATE KEYS NOT ALLOWED IN PRIMARY INDEX.

The primary index (i.e., the index through which the file is created and maintained) allows duplicate keys. This is not permitted. The program must be corrected before another attempt is made to run it.

518 <file letter> FILE: INDEX WITH DUPLICATE KEYS NOT ALLOWED FOR SEC. FILE.

An attempt has been made to access a secondary file via an index that allows duplicate keys. This in not permitted. The program must be corrected before another attempt is made to run it.

519 <file letter> FILE: FILE CONTAINING CONTROL BREAK FIELD WAS NOT READ.

The source of a report control break field is a file which has not been read. In order to ensure that the file is read, its access should be critical. The error implies that the data files are not complete; a record is missing.

520 INVALID CONTROL BREAK FIELD.

The source of a report control break field can only be a file buffer (i.e., file code A-I) or a P Register. The program must be corrected before another attempt is made to run it.

521 INVALID CONTROL BREAK FIELD.

The source of a report control break field, when specified in terms of an offset and length, can only be a file buffer (i.e., file code A-I). The program should be corrected before another attempt is made to run it.

522 <field number> FIELD: SOURCE FIELD NOT INCLUDED IN RECORD.

The source of this field is not included in its record. Any field that is referenced as a source must have INCLUDE IN RECORD set to Y. The program should be corrected before another attempt is made to run it.

523 ENTRIES OPEN/CLOSE ERROR.

The indicated file cannot be created due to an ISAM error. Try initializing or reloading the B20 system software. The operating system error code may provide an indication of the nature of the problem.

524 <file letter> FILE: INVALID INDEX NUMBER.

An attempt has been made to open the indicated file via a nonexistent index. Add the specified index to the file or change the index number to be used before attempting to run this program again. This error can also result from not defining the key for the specified index.

525 <file letter> FILE: INDEX WITH DUPLICATE KEYS NOT ALLOWED FOR SEC. FILE.

An attempt has been made to access a secondary file via an index that allows duplicate keys. A file can only be accessed as a secondary file via an index with unique keys. Correct the problem before attempting to run the program again.

526 B20 SYSTEM ERROR.

A forms run-time software error has occurred. Try initializing the B20 or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

527 B20 SYSTEM ERROR.

An error has occurred in the B20 video-access software. Try initializing the B20 or reloading the system software. The operating system error code may provide an indication of the nature of the problem.

528 <field number> FIELD: NAME REQUIRED FOR INQUIRY FIELDS.

This inquiry program has a field with no name. This is illegal for inquiries. The program must be corrected before another attempt is made to run it.

529 INVALID CONTROL BREAK FIELD.

A Data Manager error has occurred in the control break mechanism. Your software supplier should be notified of the problem.

530 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

An undeclared file has been referenced as a source. The program should be corrected before another attempt is made to run it.

531 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

An undeclared file has been referenced as a source. The program should be corrected before another attempt is made to run it.

532 <field number> FIELD: SOURCE PARAMETER IS INCORRECT.

An invalid source code has been referenced. The program should be corrected before another attempt is made to run it.

533 <range specification number> RANGE SPEC: FIELD NOT INCLUDED IN RECORD.

A field declared as part of the lower key of a range of records is not included in the record. This type of field must be marked as included in the record. The program should be corrected before another attempt is made to run it.

534 KEY FOR RANGE OF RECORDS LONGER THAN 64 BYTES.

The fields specified as the lower key in a range-of-records specification total more than the limit of 64 bytes for an ISAM key. The program should be corrected before another attempt is made to run it.

535 <range specification number> RANGE SPEC: FIELD NOT INCLUDED IN RECORD.

A field specified as part of the upper key of a range of records in not included in the record. To be accessible, all such fields must be included in the record. The program should be corrected before another attempt is made to run it.

536 KEY FOR RANGE OF RECORDS LONGER THAN 64 BYTES.

The fields specified as the upper key in a range-of-records specification total more than the limit of 64 bytes for an ISAM key. The program should be corrected before another attempt is made to run it.

537 <report item number> ITEM: INVALID TOTAL REGISTER NUMBER.

There are only 16 totals. A given total reference must therefore be in the range 1-16. The program should be corrected before another attempt is made to run it.

538 <field number> FIELD: INCORRECT SYSTEM REGISTER REFERENCE.

There are only 23 system registers (X Registers). Reference to a system register must therefore be in the range 1-23. Note also that system registers 3, 13-19, and 22 are not valid for this program type.

539 <statement number> STATEMENT: REFERENCE TO UNDECLARED SECONDARY FILE.

An attempt has been made to reference a secondary file that has not been declared. The program should be corrected before another attempt is made to run it.

540 <statement number> STATEMENT: REFERENCE TO UNDECLARED SECONDARY FILE.

An attempt has been made to reference a secondary file that has not been declared. The program should be corrected before another attempt is made to run it.

541 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

An attempt has been made to reference an invalid program register. Note that there are only 45 program registers. These registers are numbered P1-P45. The program should be corrected before another attempt is made to run it.

542 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

An attempt has been made to reference a n invalid total register. Note that there are only 16 total registers. These registers are numbered T1-T16. The program should be corrected before another attempt is made to run it.

543 INVALID NUMERIC VALUE.

An invalid numeric field has been encountered. Note that numeric fields can consist only of the digits 0-9 and a sign.

544 <number> FIELD/STATEMENT/ITEM: ARITHMETIC OVERFLOW.

An attempt has been made to assign too large a value to a field of the type integer. An integer field can contain only numbers from -9999 to +9999.

545 INVALID NUMERIC VALUE.

An invalid numeric field has been encountered. Numeric data can consist only of the digits 0-9 and a sign, if necessary.

546 INVALID NUMERIC VALUE.

An invalid numeric field has been encountered. Numeric data can consist only of the digits 0-9 and a sign, if necessary.

547 PRIMARY FILE KEY MUST HAVE AT LEAST ONE KEYBOARD ENTRY.

A Maintenance program must have at least one keyboard-entered field within its primary key. In order to satisfy this requirement, you may need to declare a dummy field; i.e., a field not included in the record but with keyboard entry.

548 <file letter> FILE: INVALID KEY SPECIFICATION.

A nonexistent field has been specified as a key field. The program should be corrected before another attempt is made to run it.

549 <file letter> FILE: THIS FILE ALREADY CONTAINS RECORD TO BE ADDED.

A duplicate key has been found while adding records to a secondary file using BATCH UPDATE. Either remove the old file first or make sure that duplicate records are not added to the file.

550 LIBRARY VERSION INCOMPATIBLE.

A library selected externally has the wrong version number for this release of Data Manager. If necessary, the library can be upgraded using DMUPGRADE.

551 <statement number> STATEMENT: INVALID LINE NUMBER IN GOTO STATEMENT.

The indicated statement contains a GOTO instruction that references an invalid line number. The program should be corrected before another attempt is made to run it.

552 <statement number> STATEMENT: GOTO BEYOND END OF FUNCTION.

The indicated statement contains a GOTO instruction that references a line number greater than the number of lines in the function. The program should be corrected before another attempt is made to run it.

553 DMRDATA FILE ERROR.

The DMRUN system file ia corrupt or incomplete. Restore this file from backup before continuing.

554 DISK FULL - CANNOT SPOOL OR PRINT TO A DISK FILE.

There is insufficient space on disk to store the spool file or disk file. Remove any unwanted files before attempting to run this program again.

555 THERE IS NO KEYBOARD ENTRY BEYOND THE KEY.

A Maintenance program must include at least one keyboard-entered field beyond the first key, as well as at least one keyboard-entered field within the first key. A Real-time Update program must include at least one keyboard-entered field.

556 <file letter> FILE: INVALID SECONDARY FILE SPECIFICATION.

The specification of the indicated secondary file does not contain the name of a record description program. The program must be corrected before it can be successfully run.

558 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement contains a reference to a non-existent field. Correct the program before another attempt is made to run it.

559 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement contains a reference to an invalid program register. Correct the program before another attempt is made to run it.

560 <statement number> STATEMENT: VARIABLE NOT RECOGNIZED.

The indicated statement contains a reference to an invalid report register. Correct the program before another attempt is made to run it.

561 <number> SORT SPEC: INVALID SORT SPECIFICATION.

The indicated sort specification is invalid. It must reference a valid field. Correct the program before another attempt is made to run it.

562 <file letter> FILE: FILE MUST BE UPDATE OR ADD FOR DELETE TO BE ALLOWED.

A secondary file must be declared with Access Control 3 (update) or 4 (add) in order to be able to delete records from it. Secondary files with Access Control set to 1 or 2 are opened for input only and cannot be altered in any way.

563 <file letter> FILE: FILE MUST BE UPDATE OR ADD FOR DELETE TO BE ALLOWED.

A secondary file must be declared with Access Control 3 (update) or 4 (add) in order to be able to delete records from it. Secondary files with Access Control set to 1 or 2 are opened for input only and cannot be altered in any way.

564 <file letter> FILE: FILE MUST BE UPDATE OR ADD FOR DELETE TO BE ALLOWED.

A secondary file must be declared with Access Control 3 (update) or 4 (add) in order to be able to delete records from it. Secondary files with Access Control set to 1 or 2 are opened for input only and cannot be altered in any way.

565 ATTEMPT TO DELETE RECORD FROM NON-EXISTENT TRANSACTION LOG.

A "DELETE A" instruction has been performed in the update function, but there is no A file (i.e., a transaction log). You must correct the program before it can successfully be run.

566 <number> FIELD/STATEMENT/ITEM: ARITHMETIC OVERFLOW.

An arithmetic function involving an extremely large real number (i.e., a number greater than 10 to the 15th power) has resulted in an arithmetic overflow. The program should be written so as to exclude numbers this large.

567 <number> FIELD/STATEMENT/ITEM: ARITHMETIC UNDERFLOW.

An arithmetic function involving an extremely small real number (i.e., a number smaller than 10 to the -15th power) has resulted in an arithmetic underflow. The program should be written so as to exclude numbers this small.

APPENDIX D

DMPLIST ERROR MESSAGES

1 CANNOT READ CONTROL RECORD IN DMRDATA.

Note that this error message cannot be translated.

Reload DMRDATA from your release disk to the <DM> directory.

2 DMRDATA INCOMPATIBLE WITH THIS VERSION OF DMPLIST.

Note that this error message cannot be translated.

Reload DMRDATA from your release disk to the <DM> directory.

3 DMPDATA INCOMPATIBLE WITH THIS VERSION OF DMPLIST.

Reload DMPDATA from your release disk to the <DM> directory.

4 DM PROGRAM CORRUPT: PROGRAM UNUSABLE.

This indicates a Data Manager failure and should be reported to your software supplier.

5 DMRDATA FILE ERROR.

DMPLIST could not close DMRDATA.

6 CANNOT OPEN <DM>DMRDATA.

Note that this error message cannot be translated.

DAM error when opening DMRDATA. Reload DMRDATA from your Data Manager release disks to the <DM> directory.

7 ADDITIONAL MEMORY REQUIRED.

Insufficient memory. The host B20 must have a minimum of 256KB of memory. Be sure that memory is not being wasted by an unnecessary installation of the spooler or an unnecessarily large ISAM. The operating system error code may provide additional information.

8 DM MEMORY MANAGEMENT ERROR.

A problem exists in DMPLIST that affects the management of memory. Please file a trouble report with your software supplier.

9 B20 SYSTEM ERROR.

An error occured when de-allocating memory. Check the error code for the specific problem.

10 DM MEMORY MANAGEMENT ERROR.

A problem exists in DMPLIST that affects the management of memory. Please file a trouble report with your software supplier.

11 DMRDATA FILE ERROR.

DMRDATA record not found. Reload DMRDATA from your release disk.

12 DMRDATA FILE ERROR.

DMRDATA file error. The error code may provide an indication of the nature of the problem.

13 DMPDATA FILE ERROR.

DMPLIST could not close DMPDATA.

14 CANNOT OPEN <DM> DMPDATA.

DAM error when opening DMPDATA. Reload DMPDATA from your Data Manager release disks to the <DM> directory.

15 DMPDATA FILE ERROR.

DMPDATA record not found. Reload DMPDATA from your release disk.

16 DMPDATA FILE ERROR.

DMPDATA read error. The error code may provide an indication of the nature of the problem.

17 CANNOT SPOOL OR PRINT DIRECT - PRESS THE FINISH KEY TO EXIT.

18 CANNOT SPOOL OR PRINT DIRECT - PRESS THE FINISH KEY TO EXIT.

19 B20 SYSTEM ERROR.

Cannot open print file.

20 B20 SYSTEM ERROR.

Cannot open print file.

21 PROGRAM LIBRARY FILE ERROR.

Error in CloseIsam.

22 ISAM NOT INSTALLED. INSTALL ISAM BEFORE CONTINUING.

23 PROGRAM LIBRARY FILE ERROR.

Error in OpenIsam. Error code can apply to either data file or index file.

24 PROGRAM LIBRARY FILE ERROR.

ISAM error while attempting to load program from Program Library into memory.

25 PROGRAM LIBRARY FILE ERROR.

ISAM error while attempting to read Program Library control record.

26 LIBRARY VERSION INCOMPATIBLE.

27 B20 SYSTEM ERROR.

VAM error while writing error message.

28 B20 SYSTEM ERROR.

Beep (to sound alarm) returned an error.

29 B20 SYSTEM ERROR.

Error in GetDateTime.

30 B20 SYSTEM ERROR.

Error in ExpandDateTime.

31 B20 SYSTEM ERROR.

Error when encoding system date.

32 B20 SYSTEM ERROR.

Error when encoding system date.

33 B20 SYSTEM ERROR.

Error when encoding system date.

34 INVALID SYSTEM DATE CODE IN SYSTEM FILE CONTROL INFORMATION.

The control record in DMRDATA has been incorrectly modified.

35 B20 SYSTEM ERROR.

Error in GetDateTime.

36 B20 SYSTEM ERROR.

Error in ExpandDateTime.

37 B20 SYSTEM ERROR.

Error when encoding system time.

38 B20 SYSTEM ERROR.

Error when encoding system time.

39 B20 SYSTEM ERROR.

Error when encoding system time.

40 PROGRAM LIBRARY FILE ERROR.

ISAM error while attempting to load program from Program Library into memory.

41 PROGRAM LIBRARY FILE ERROR.

ISAM error while attempting to load program from Program Library into memory.

42 DM PROGRAM NOT FOUND OR INCOMPLETE.

An attempt has been made to load a nonexistent or incomplete program. Check the directory to make sure that the program is present or be sure that the name of the requested program is correct.

43 PROGRAM VERSION INCOMPATIBLE.

44 ADDITIONAL MEMORY REQUIRED TO CONTINUE THIS PROGRAM.

There is insufficient memory available to load this program. Create a simpler program with lower memory requirements.

45 B20 SYSTEM ERROR.

Error in QueryMemAvail.

46 DM APPLICATION PROGRAM CORRUPT: PROGRAM UNUSABLE.

The program being loaded has an invalid program type code. This could be caused by a corrupt program library. Try replacing the program library with a backup copy.

47 DATA MANAGER SYSTEM ERROR.

DMPLIST tried to access a nonexistent LED.

48 CANNOT OPEN <DM>DMRFORMS.

Reload DMRFORMS from your release disk to the <DM> directory.

49 DMRFORMS FILE ERROR.

DMPLIST could not close DMRFORMS.

50 DMRFORMS FILE ERROR.

DMPLIST could not access a form in DMRFORMS. Reload DMRFORMS from your Data Manager release disk.

51 DMRFORMS FILE ERROR.

A form used by DMPLIST has been incorrectly modified in DMRFORMS.

52 B20 SYSTEM ERROR.

FormsRuntime has returned an unexpected error. It could be either a system-software or Data Manager problem.

53 THIS ENTRY MUST BE ALL NUMBERS.

54 DMRFORMS FILE ERROR.

A form used by DMPLIST has been incorrectly modified in DMRFORMS.

55 DMRFORMS FILE ERROR.

A form used by DMPLIST has been incorrectly modified in DMRFORMS.

56 DMRFORMS FILE ERROR.

A form used by DMPLIST has been incorrectly modified in DMRFORMS.

57 DATA MANAGER SYSTEM ERROR.

DMPLIST unable to allocate memory to store form while attempting to display help form.

58 B20 SYSTEM ERROR.

Unable to restore form after help form is displayed.

59 B20 SYSTEM ERROR.

Unable to de-allocate memory after displaying help.

60 DMRFORMS FILE ERROR.

A DMPLIST form has been incorrectly modified in DMRFORMS.

61 DMRFORMS FILE ERROR.

A DMPLIST help form missing from DMRFORMS.

62 B20 SYSTEM ERROR.

Error in UndisplayForm when attempting to display form help.

63 B20 SYSTEM ERROR.

ResetFrame failed when attempting to display help form.

64 DMRFORMS FILE ERROR.

A form has been incorrectly modified in DMRFORMS.

65 DMRFORMS FILE ERROR.

Field help form missing from DMCHELP.

66 B20 SYSTEM ERROR.

PutFrameChars (blank out error message) returned an error.

67 THIS ENTRY MAY NOT BE BLANK. ENTER AT LEAST ONE LETTER.

68 THIS ENTRY MAY NOT BE BLANK. ENTER AT LEAST ONE LETTER.

69 YOUR SELECTION IS NOT AVAILABLE. PLEASE ENTER ANOTHER.

APPENDIX E

DMUPGRADE CONVERSION

B20 Data Manager offers the DMUPGRADE2 program for converting level-1.0 or level-1.1 programs to make them compatible with level-2.0 software. DMUPGRADE2 converts one program library at a time and lists each program on the screen after being converted.

Listed below is the procedure for converting a program library to level 2.0. Repeat this procedure for each library you wish to convert.

1. **Enter DMU and press GO.**

   To begin the program conversion procedure, enter the command DMU and press GO.

   The DMUPGRADE2 Program screen will be displayed.

2. **Enter library name.**

   Enter the name of the program library you wish to convert to level 2.0. DMUPGRADE2 will convert each program in that library and then display the following message.

   ```
   Converting <program name> Done.
   ```

   After all programs in the library have been converted, the DMUPGRADE2 program will go to End-of-Job.

   The original library will be renamed by attaching "-Old" at the end of the name, and the upgraded library will assume the original library name. For example, if you convert a level-1.0 library called TESTLIB, the level-1.0 library will be renamed TESTLIB-Old, and the level-2.0 library will be called TESTLIB.

INDEX

DOCUMENTATION EVALUATION FORM

Title of Manual: B20 Data Manager Programmer's Manual

Form Number: 1154333

Date of Publication: May 1983

Burroughs Corporation is interested in receiving your comments and suggestions regarding this manual. Comments will be used to improve this manual when it is revised.

Please limit yourself to one suggestion per form. Make copies of this form if you have additional suggestions. Please circle the type of your suggestion from the list below:

ADDITION DELETION REVISION ERROR

COMMENTS (regarding page _____):

____

____

____

____

From:

Name ______________________________________

Title ______________________________________

Company ______________________________________

Address ______________________________________

Phone Number ____________ Date __________

Remove form and mail to:

Documentation Department
Systems Development Group
Burroughs Corporation
3519 West Warner Avenue
Santa Ana, California 92704