WANG

VS

File Management Utilities Reference

VS
File Management Utilities Reference

1st Edition — December, 1981

800-1308FM-01

WANG LABORATORIES, INC., ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL: 617/459-5000, TWX 710-343-6769, TELEX 94-7421

Disclaimer of Warranties and Limitation of Liabilities

The staff of Wang Laboratories, Inc., has taken due care in preparing this manual; however, nothing contained herein modifies or alters in any way the standard terms and conditions of the Wang purchase, lease, or license agreement by which this software package was acquired, nor increases in any way Wang's liability to the customer. In no event shall Wang Laboratories, Inc., or its subsidiaries be liable for incidental or consequential damages in connection with or arising from the use of the software package, the accompanying manual, or any related materials.

NOTICE:

All Wang Program Products are licensed to customers in accordance with the terms and conditions of the Wang Laboratories, Inc. Standard Program Products License; no ownership of Wang Software is transferred and any use beyond the terms of the aforesaid License, without the written authorization of Wang Laboratories, Inc., is prohibited.

This manual replaces the discussions of the File Management Utilities in the *VS Utilities Reference* (800-1303UT-02). For specific changes made to the utility programs and/or their descriptions, refer to the "Summary of Changes".

WANG LABORATORIES, INC., ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL: 617/459-5000, TWX 710-343-6769, TELEX 94-7421

PREFACE

Wang provides a variety of programs with the VS Operating System that perform routine data processing operations. These programs, referred to as utilities, extend the Operating System support of the programming environment. The VS utilities are documented in a set of three manuals, each describing a different phase of the data processing environment. Although any one of the manuals can be used separately, their combined use is often appropriate.

VS Program Development Tools - 800-1107PT
Documents the program development tools provided with the VS Operating System. The VS Operating System includes a program development EDITOR through which the user can enter, modify, compile, and/or run an application program. Compiled or assembled program modules can then be linked through the LINKER utility. The VS also provides a Symbolic Debugger, through which the user can monitor program processing.

VS File Management Utilities Reference - 800-1300FM
Documents the utilities that facilitate the data entry and retrieval process. Data files can be constructed, modified, and reported on through the VS File Management Utilities. All File Management Utilities require a description of the data file that is stored in a control file.

VS System Utilities Reference - 800-1303UT
Documents utilities that perform the support tasks involved in program and file processing. Files can be transferred from one storage medium to another, files can be viewed on the workstation or on a printout, files can be sorted, and system storage media can be initialized or examined for accuracy.

This manual describes the VS File Management Utilities. These programs perform the tasks of creating, maintaining, and reporting on data files. Each of these utilities requires that the data file or view being processed have an associated control file containing information regarding the actual data records (such as file size, field names, and record length).

The programs described in this manual can be run by programmers and nonprogrammers alike; however, a rudimentary knowledge of at least one programming language is recommended. This manual assumes familiarity with topics discussed in the following manuals.

VS Programmer's Introduction	800-1101PI

In addition, topics treated in the following manuals provide additional information that may be helpful when used in conjunction with the discussions of individual File Management Utilities.

VS Operating System Services	800-1107OS
VS Principles of Operation	800-1100PO
VS System Operation Guide	800-1102SO
VS System Management Guide	800-1104SM
VS Procedure Language Reference	800-1205PR
VS Assembler Language Reference	800-1200AS
VS BASIC Language Reference	800-1202BA
VS COBOL Reference	800-1201CB
VS FORTRAN Reference	800-1208FR
VS RPG II Language Reference	800-1203RP
VS Useraids Reference	800-1301UA

SUMMARY OF CHANGES

The chapters contained in this manual were previously published in the VS Utilities Reference (800-1303UT-02). The list below summarizes changes to the documentation since its last publication.

Utility	Description of Change	Affected Pages
CONDENSE	Chapter completely rewritten.	7-1 to 7-15
CONTROL	Chapter includes minor rewrites, as well as User Exit documentation.	2-1 to 2-20
DATENTRY	The following software changes are described. Options are now available to specify 1) either upper/lower case entries for character fields or uppercase entries only, or 2) field display characteristics for data entry (underlining and field intensity).	3-1 to 3-4
EZFORMAT	The Data Entry option description has been rewritten. All other options of EZFORMAT are described in the VS System Utilities Reference.	4-1 to 4-14
INQUIRY	Chapter completely rewritten.	6-1 to 6-9
REPORT	Chapter includes minor rewrites and a completely rewritten section on User Exits.	5-1 to 5-21
Introduction	Completely rewritten.	1-1 to 1-5
Appendices	Miscellaneous technical and editorial changes.	

CONTENTS

FIGURES

TABLES

CHAPTER 1
INTRODUCTION TO THE FILE MANAGEMENT UTILITIES

1.1 OVERVIEW

The Wang VS provides a group of system utility programs that enables a user with relatively limited data processing experience to define, access, inspect, and report on one or two DMS data files. Collectively, these utilities are known as the File Management Utilities. The functions provided by each of these utilities are summarized as follows.

CONTROL — Creates a control file that defines all field, record, and file attributes for a specified data file. Each record in a control file defines several aspects of a field in its corresponding data file. Once the parameters have been specified through CONTROL, the data file itself can be easily built and maintained with the File Management Utilities DATENTRY or EZFORMAT. Refer to Chapter 2.

DATENTRY — Lets the user create or maintain records in a data file. Each field is defined according to the specifications in a corresponding control file. DATENTRY also lists on the screen or prints out file records for easy reference. Refer to Chapter 3.

EZFORMAT — Generates a customized data entry and maintenance program corresponding to a user-designed screen format and a control file. EZFORMAT provides an alternative to DATENTRY for existing DMS files. Refer to Chapter 4. (The other functions of EZFORMAT are described in the VS System Utilities Reference.)

REPORT — Creates a Report Definition file (based on the information in a control file) that defines the parameters and format of a printed report, using the contents of a data file as input. Refer to Chapter 5.

INQUIRY — Interrogates and tests a data file for user-specified field values. Refer to Chapter 6.

CONDENSE — Creates a data file with a single record type from a file with multiple record types, so that a report based on the file's data can be generated. A control file corresponding to the output data file is also constructed. Refer to Chapter 7.

1.2 DATA PROCESSING WITH THE FILE MANAGEMENT UTILITIES

For processing files through the File Management Utilities, the user works with one, and sometimes two, data files at a time. To process a data file, the file must be defined and then created. Once the data file exists, the user can process reports or inquiries of the data file. The use of the File Management Utilities for data file processing is illustrated in Figure 1-1. In addition, the user can change a data file containing multiple record types into one containing one record type. The resulting file can then be accessed through the File Management Utilities.

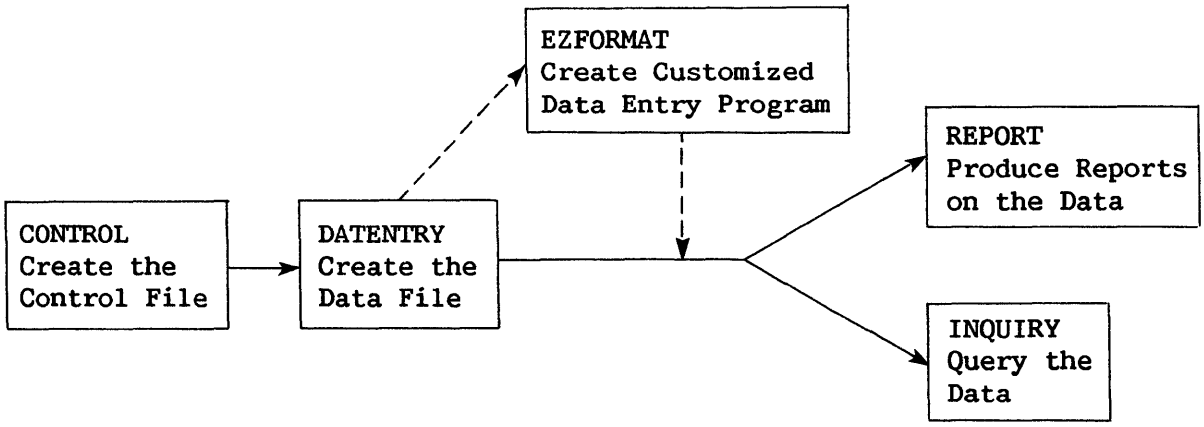


Figure 1-1. Data Processing with the File Management Utilities

A data file is defined through the File Management Utilities with the CONTROL utility. Through CONTROL, the user defines the physical and logical structure of the data file. This information includes general information about the data file, such as file organization, as well as the characteristics of the individual fields, such as name, length, and internal format. Additionally, validation criteria for data entry can be specified. These definitions are stored in a file known as a control file. Each control file normally defines only one data file, but each data file may have an arbitrary number of control files associated with it. (Multiple control files enable different users to access portions of the same data file.) Once the characteristics of the data file have been defined by creating the control file, the user is ready to create the actual data file.

A data file is created through the File Management Utilities by using the DATENTRY utility. When creating a data file, a control file is used to provide the format of the data file. Data entered into the new data file is validated using the criteria specified in the control file. If acceptable, the data is written into the new, empty data file. Once the data file has been created, the user can add, delete, or modify entered data through DATENTRY. Again, the control file provides formatting and validation specifications for the data.

If the user prefers to use a customized screen format for data entry instead of the default screen supplied by the DATENTRY utility, the EZFORMAT utility can be used. Through EZFORMAT, the user defines a data entry screen that is incorporated into a data entry program. The resulting program offers options for data addition, modification, and deletion using the customized screen format. Data file creation, however, cannot be performed through the customized data entry program.

To report on one or two data files through the File Management Utilities, the REPORT utility is used. REPORT offers a variety of options that enables the user to specify the contents and format of a report. In addition, new fields can be defined and summary data can be specified. The report definition is permanently saved in a report definition file, so that in addition to generating routine reports, it can be periodically reviewed and revised according to changing needs. Reports can either be printed or examined on the screen.

A data file can be interrogated using the INQUIRY utility. Selected portions of the data file can be retrieved by means of queries formulated in conversational English. The retrieved data can be displayed on the workstation screen, or can be stored in an output file for use in a report at a later time.

In addition to the File Management Utilities shown in Figure 1-1, a utility program is provided that enables the user to process data in a file containing multiple record types. This utility, known as CONDENSE, creates a new data file with only one record type from the original data file with multiple record types. CONDENSE also creates a corresponding control file for the new data file. Once a data file has been processed with CONDENSE, its data can be accessed through DATENTRY, EZFORMAT, REPORT, and INQUIRY, and its control file can be modified through certain CONTROL options.

1.3 UTILITIES IN THE VS ENVIRONMENT

All VS File Management Utilities reside in the @SYSTEM@ library on the System Program volume. The file name of each utility is the name of the utility; thus, the DATENTRY utility is located in the file DATENTRY in the @SYSTEM@ library on the System Program volume.

1.3.1 Running Utilities

The user can run utilities either directly through the Command Processor or under procedure control. If the utility is run through PF1 of the Command Processor, the user need only enter the utility's name in the file parameter. The library and volume names can be omitted (or default values left unchanged) because the VS always searches the @SYSTEM@ library on the System Program volume if the specified file name cannot be located in the indicated library and volume location. Note that the default library should be erased if the default library contains a file with the utility's file name. If the utility is run under procedure control, the user need only specify the utility name following the Procedure language RUN statement; i.e., RUN DATENTRY.

1.3.2 Utilities and the VS Procedure Language

The VS File Management Utilities are designed so that workstation interaction can be controlled or minimized through the VS Procedure language. Thus, most utility information requests are processed through GETPARMs, which are the VS Procedure language interface to a program. A brief description of GETPARMs and the VS Procedure language can be found in Appendix A, File Management Utility GETPARMs. In addition, the description of each utility in this manual terminates with a sample procedure to guide the user in writing customized procedures.

CHAPTER 2
CONTROL

2.1 INTRODUCTION

The CONTROL utility can be used to define the attributes of a data file. The file and record attributes are placed in a control file that serves as a directory to describe data file attributes.

Any data file or data base to be created or modified through DATENTRY, maintained by an EZFORMAT-generated data entry program, or accessed by REPORT, INQUIRY, or CONDENSE, must have at least one associated control file. A control file consists of one file descriptor record, a number of field descriptor records (one for each field in the data file), one or two alternate key records, and, optionally, one to three comment records. The file descriptor record consists of fields that define the data file at the file level (e.g., total record length). The field descriptor records consist of fields that define the data file at the field level (field attributes and field validation criteria).

CONTROL allows the user to impose limits on data input, update, and output through the DATENTRY utility. In this way, the accuracy of the data can be assured, constraints can be placed on field updating, and data can be reported simply and accurately. The control file is the basis of this data control function.

Although the control file contains a complete description of a data file, the data file and its corresponding control file exist independently of each other. Thus, while control files can be readily modified, care must be taken to ensure that any modifications do not affect the structure of the data file. If a future data file expansion or change is anticipated, it may be useful to reserve extra space in the data file when the control file is originally created. For example, fields could then be added to the control file at a later date without destroying the existing data file structure. However, the existing data file is not automatically updated. When DATENTRY is run following a control file change, the new field is available for entry if new records are added to the file. Existing records remain unchanged until the user retrieves them for modification, at which time the user is able to enter data into the new field. Unless appropriate space is provided, control file changes that affect the length of the data record require creation of a new data file. This is also true when the internal format of a given field is changed, affecting the field's length. The CREATE user aid or a user-written program may be used to restructure an existing data file.

An overview of CONTROL processing is provided in Figure 2-1.

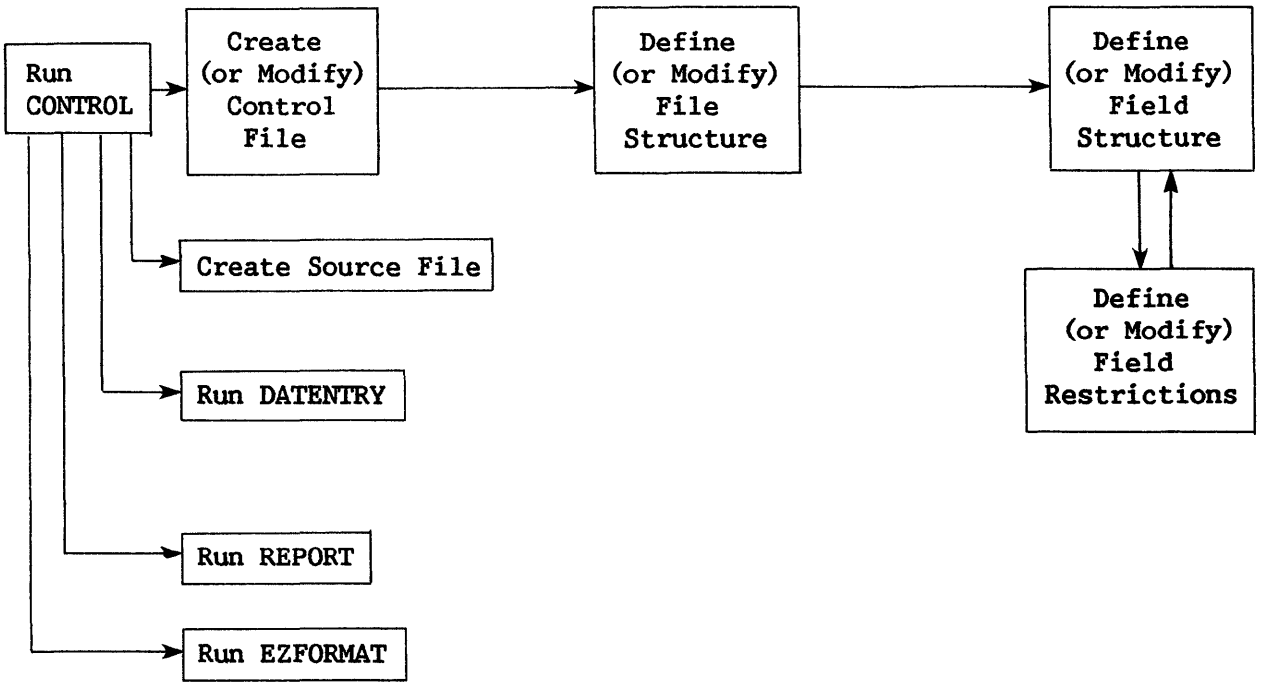


Figure 2-1. CONTROL Processing

2.2 RUNNING THE CONTROL UTILITY

The first CONTROL display requests the name and location of a control file. Specify either an existing file to be modified or examined or a new one to be created. The name supplied is used as the control file name. A default library name for the control file is created by concatenating the User ID with the letters CTL. If a default input volume name has been specified with the SET command, it is supplied as the volume for the control file. The control file name must be from 1 to 8 characters in length and may not contain embedded blanks. CONTROL processing can be terminated from this display by pressing PF16.

Once the control file is identified, CONTROL displays a menu offering the following options; each option is documented in the indicated section or chapter of this manual.

PF Key	Option	Section or Chapter
2	Create a Control File allows the user to create a new control file with the name given on the first screen.	2.3
3	Add Records to a Control File allows the user to add records to an existing control file.	2.4.1
4	Modify Control File Header or Field Records allows the user to make changes in either the general data file characteristics specified in the control file header or in specific field characteristics.	2.4.2
5	Delete Records from a Control File allows the user to delete records from an existing control file.	2.4.3
6	List Records on a Control File allows the user to view or print the contents of the control file.	2.4.4
7	Maintain Table Entries allows the user to create a table file or list, add, delete, or change the contents of data validation tables created by the CONTROL utility.	2.4.5
8	Create a Source File from This Control File creates a source file defining the data file described by the control file. Two source languages are available: COBOL and RPG II.	2.6
9	Run Wang VS Data Entry Utility allows the addition, modification, or deletion of information in a data file described by a control file.	3
10	Run Wang VS Report Utility allows a report to be created from up to two data files described by control files.	5
11	Run Wang VS Screen Formatting Utility allows a screen display to be defined through EZFORMAT.	4
12	Modify Field Update Sequence permits the user to selectively rearrange the sequence of specified data fields appearing on the DATENTRY display screen.	2.4.6
16	Exit to Respecify Control File Location restarts the CONTROL utility at the DEFINE CONTROL FILE screen, thus allowing the utility to be run again or ended.	

The user can impose restrictions on a field in a data file in addition to those supported by CONTROL. Section 2.3.1, Data File Parameters, and Section 2.5, User Exit Subroutines, describe the process of extending field restrictions. Section 2.8, Sample CONTROL Procedure, documents the process of controlling user interaction with the CONTROL utility.

2.3 CREATING A CONTROL FILE

To create a new control file for a data file, select PF2 from the Control File Options screen. Two types of information must be specified: the parameters for the data file, and the field specifications for individual records within the data file. Refer to Appendix B, Control File Record Formats, for a description of the control file record formats.

2.3.1 Data File Parameters

The file descriptor (header) record of the control file is built by entering values for the data file attributes into the Data File Parameters display, which appears when PF2 is selected from the Control File Utility menu. In this way, the attributes of the data file are designated. The data file attributes are listed as follows.

- Record Length
- Key Field Name
- Alternate Key Fields
- User Exit
- Report Code
- Update Code
- Delete Code
- File Type
- Comments

Record Length -- The data file record length (RECLEN) may range from 1 to 2040 bytes for indexed fixed length records, from 1 to 2048 bytes for consecutive fixed length records, and from 1 to 2024 bytes for variable length records and compressed length records.

The record length specified should be that expected for the data file being defined. If the records are of varying lengths, the length specified is the maximum record length. Variable length and compressed records cannot be updated or deleted when the data file organization is consecutive. However, fixed length, noncompressed records in a consecutive file can be updated but not deleted. Refer to the File Type attribute description below for more information.

Key Field Name -- If a key field is specified, an indexed data file is automatically created. The data file records, therefore, can be accessed individually through the DATENTRY Record Modification option by entering a given record's key field value. (Refer to Chapter 3, DATENTRY.)

If specified, the key field (KEYFIELD) name must be from 1 to 8 characters in length, the first of which must be alphabetic, and may not contain embedded blanks. The specified key field is considered the file's primary key. The key field cannot exceed 132 bytes in length. Duplicate primary key data values are not permitted.

Should the user decide not to specify a key field, the data file is assumed to be sequentially ordered, and a consecutive file is created.

Alternate Key Fields -- Alternate key fields may be specified only for indexed files. They operate in the same fashion as primary key fields, except that duplicate values are allowed. Alternate key fields may overlap if they are not modifiable. (For example, field KEY1 could occupy bytes 1-10, and field KEY 2 could occupy bytes 5-15.) Up to 16 alternate keys may be specified. The sum of the lengths of the primary key field plus the largest alternate key field may not exceed 255 bytes.

If alternate key fields are indicated, their names are entered into the screen appearing immediately following the completion of the data file parameters specification. At this time, the user may specify whether the individual alternate keys may have duplicate values.

User Exit -- The User Exit option permits the user to invoke a program subroutine to validate and/or perform operations on data entered though DATENTRY. A User Exit subroutine can be invoked if a subroutine name is supplied; subroutine names are limited to the values USER1 through USER10. The procedure for generating and using a User Exit subroutine is discussed in Section 2.5.

Report Code -- The file report code indicates whether or not the REPORT utility is permitted to access the data file. A value of 0 allows reporting, while a value of 1 prohibits it.

Update Code -- The file update code indicates whether or not the DATENTRY utility may be used to add, delete, or change records in the data file. A value of 0 allows updating, while a value of 1 prohibits it.

Delete Code -- The delete code indicates whether or not the DATENTRY utility may be used to delete records from the data file. A value of 0 allows record deletions, while a value of 1 prohibits them. Only records in indexed files may be deleted. To delete records in consecutive files, use the CREATE user aid.

File Type -- The file type may be either F, V, or C to indicate, respectively, whether the data file contains fixed length records, variable length records, or compressed records. F is the default.

A file type of V sets a flag on the data file indicating that it contains more than one type of record format. Since DATENTRY cannot create a data file containing more than one type of format, a file type of V should be set only when the user wishes to enter data by some means other than through DATENTRY. DATENTRY cannot modify data in a consecutive data file containing variable length or compressed records. If type V is selected, the DATENTRY utility will create a format such that all records are the length of their longest entry. The DATENTRY functions possible for the file type and record length combinations are summarized in Table 2-1.

Table 2-1. Effect of File Type on DATENTRY Operation

DATENTRY Option	Consecutive, Fixed Length	Consecutive, Compressed or Variable Length	Indexed, Fixed Length	Indexed, Compressed or Variable Length
Add records	Allowed	Allowed	Allowed	Allowed
Delete records	Not allowed	Not allowed	Allowed	Allowed
Modify records	Allowed	Not allowed	Allowed	Allowed

Comments

For the purpose of internal documentation only, up to 3 comment records may be entered. In this way, a user can make a notation that appears when the records in the control file are listed through PF6 on the Control File Utility menu.

2.3.2 Data Field Definitions

The data field definitions specify the attributes of each field in a record in the data file. Information specified for each field definition is stored in a separate field descriptor record in the control file. The data field attributes are specified on the Data Field Definitions screen, which is first displayed following the Data File Definitions screen and subsequently displayed when each previous field definition is complete. When all fields have been defined, field definition can be terminated from the Data Field Definitions display by pressing PF16. The data field attributes are listed as follows.

- Field Name
- Starting Location
- Internal Format
- Internal Length
- External Length
- Decimal Positions
- Occurrences
- Report Code
- Update Code
- Display Code
- Zero Suppress
- Sign Control
- Dollar/Comma
- Binary Edit
- Date Stamp
- Cumulative Field
- Field Alias

Field Name -- The entered name is used as the field name, and refers to the same field from any of the File Management utilities. The name appears as a prompt when entering field values through DATENTRY. It is also used within the data file for INQUIRY and CONDENSE, as well as for the default column heading used on reports generated by REPORT. The field name must be from 1 to 8 characters in length, the first of which must be alphabetic. The field name may not contain embedded blanks.

Starting Location -- The starting location is the byte number within the data record at which the field is to start. Valid starting locations range from the first byte in the record (1) to the data file record length as defined in the data file specifications.

When a field descriptor record is created, the user must specify the starting location of the first byte of the first field. Subsequently, the system assesses the internal length attribute of the field and automatically provides the location of the next open byte as the starting location for the next Data Field Definitions display. This system-supplied starting location can be overridden, if the user wishes, to permit specification of fields in any arbitrary order. Note, however, that accumulator fields must be specified before their source fields. (Refer to Cumulative Field in this section.) The starting location must be an integer and may not exceed the record length. Nonmodifiable fields may overlap each other or modifiable fields, but two modifiable fields may not overlap. Note that overlapping fields contain common data and are equivalent to a COBOL group item. Overlapping fields are designed to be used as key fields.

Internal Format -- The internal format defines how the field is to be physically stored on disk. There are five permissible internal formats: C (character), B (binary), Z (zoned decimal), U (unsigned), and P (packed decimal). Character fields must have an internal format of C; they are stored in ASCII code (one character per byte) and may be up to 67 bytes long if modifiable (132 bytes if nonmodifiable). Numeric fields must have an internal format of B, Z, U, or P. Binary fields may be represented externally as either decimal (CVD) or hexadecimal (HEX), whichever is indicated by the binary edit code. Binary CVD format allows entry of numeric values (0-9) only, while binary HEX format allows the hexadecimal digits A, B, C, D, E, and F to be entered, as well as 0 through 9. Internally, binary fields may not exceed 4 bytes, packed fields may not exceed 8 bytes, and zoned and unsigned fields may not exceed 15 bytes. For further discussion of the zoned, unsigned, and packed decimal formats, refer to Chapter 7, A Sample Application Using CONTROL, DATENTRY, and REPORT; Appendix D, Data Formats; and the VS Principles of Operation. This information is summarized in Table 2-2.

Table 2-2. Internal Format Description

Internal Format	Maximum Internal Length	Internal Length Calculation	Valid Entry Characters
C	67 or 132 bytes	1 char/byte	all alphanumerics
B (CVD)	4 bytes	Refer to Table 2-4	0-9, "-"
B (HEX)	4 bytes	2 char/byte	0-9, A-F
Z	15 bytes	1 byte/digit	0-9, "-", "."
U	15 bytes	1 byte/digit	0-9, "."
P	8 bytes	1 byte for first digit and sign; 2 digits/byte thereafter	0-9, "-", "."

Internal Length -- The internal length is a positive integer specifying how many bytes to reserve for the field in the data record. The field may not run off the end of the record; that is, the starting location plus the internal length minus one must not exceed the record length. If the update codes indicate the data field may be updated by DATENTRY, CONTROL checks to ensure that the field does not overlap other updatable fields (refer to Starting Location in this section). Modifiable fields are limited to 67 bytes. If the field is not modifiable, it may overlap other non-modifiable fields. Note that overlapping fields, or portions of fields, contain the same data. Consult Table 2-2 for maximum internal length values, and Appendix D, Data Formats.

External Length -- The external length is a positive integer that specifies how many character or digit positions are used when the field is represented externally by DATENTRY. It is not possible to specify an external length larger than the values listed in Tables 2-3 or 2-4. If the external field length is left blank (the default action), the external length is automatically calculated according to the algorithms in Tables 2-3 and 2-4. User specification of external length overrides this automatic calculation. Refer to Appendix D, Data Formats, for further information.

Table 2-3. Default External Length Values

Internal Format	External Length Calculation (in bytes)
C	Internal length
B (decimal - CVD)	Refer to Table 2-4
B (hexadecimal)	2 x internal length
Z (decimal positions = 0)	Internal length + 1
Z (decimal positions ≠ 0)	Internal length + 2
U (decimal positions = 0)	Internal length
U (decimal positions ≠ 0)	Internal length + 1
P (decimal positions = 0)	2 x internal length
P (decimal positions ≠ 0)	(2 x internal length) + 1

Table 2-4. Default External Length Values for a Decimal B Format

B (decimal - CVD)

Internal Length	External Length
1	4
2	6
3	8
4	11

Decimal Positions -- The number of decimal positions specifies how many decimal places will be permitted to the right of the decimal point in a numeric field. Decimal positions can be specified only if the internal format is zoned, packed decimal, or unsigned. If specified, the decimal positions field must contain a positive integer from 0 to 9. The internal length of the field remains unchanged by the inclusion of decimal places. If the file and field update codes indicate that the field may be updated by DATENTRY, then DATENTRY ensures that the field contains the proper number of decimal positions when data is entered.

Occurrences -- The number of occurrences specifies the number of times the field is to be repeated within the data record. Occurrences must be a positive integer between 1 and 99, inclusive. If the number of occurrences is greater than 1, the field is treated as a one-dimensional array, and the field name is subscripted whenever it is represented externally by DATENTRY, REPORT, INQUIRY, or CONDENSE. If the file and field update codes indicate the field may be updated by DATENTRY, CONTROL checks to ensure the field does not overlap other updatable fields.

Report Code -- The field report code indicates whether or not the field may be accessed by REPORT. A value of 0 allows reporting, while a value of 1 prohibits it.

Update Code -- The field update code indicates whether or not DATENTRY may be used to modify this field. A value of 0 allows modification, while a value of 1 prohibits it. A field not modifiable by DATENTRY does not appear in the DATENTRY screen displays. Modifiable fields may not overlap.

Display Code -- The display code is used only if the field is updatable by DATENTRY. If the code is set to 0, the field will be cleared after the ENTER key is pressed when data is entered into the field through DATENTRY. If the code is set to 1, the data entered into the field will remain on the screen, allowing it to be entered into the next record without retyping. If the code is set to 2, data can be entered through the ADD mode of DATENTRY, but data cannot be modified once it is entered.

Zero Suppress -- The zero suppress code is used only if the field is numeric and if the file and field report codes indicate that the field may be operated on by REPORT. Zero suppression may also be selected as an option at report definition time. The three possible zero suppress codes and their effect on the external format of the field in reports generated with REPORT are summarized as follows.

Zero Suppress Code	External Format
0	No zero suppression
1	Leading (leftmost) zeros are suppressed.
2	Leading (leftmost) zeros are represented as asterisks (*).

Sign Control -- The sign control code is used only if the field is numeric and if the file and field report codes indicate that the field may be operated on by REPORT. The sign control code determines the external report format of the negative field values. If the field is lengthened as a result of adding a sign, extra space is automatically created for the field on reports. The external format corresponding to each sign control code is summarized as follows.

Sign Control Code	External Format of Negative Fields
0	No sign
1	Trailing minus sign (-)
2	Trailing CR
3	Trailing DB

Dollar/Comma -- The dollar/comma code permits the placement of dollar signs and/or commas in numeric field report formats. This option is used only if the field is numeric and if the file and field report codes indicate that the field may be operated on by REPORT. This option may also be selected through REPORT at report definition time. Extra space is automatically created in reports for the fields lengthened due to the addition of dollar signs and/or commas. Dollar signs are printed to the left of the field; commas are printed at three-digit intervals to the left of the decimal point. The external format of the field in a REPORT-generated report resulting from each dollar/comma code is illustrated as follows.

Dollar/Comma Code	External Format
0	No sign
1	Commas (,) inserted
2	Dollar sign ($) precedes field
3	Commas (,) inserted and dollar sign ($) precedes field

Binary Edit -- The binary edit code controls data entry for binary fields, and may be used only if the internal format has been defined as type B (binary). A binary edit flag of 0 indicates that the field, which includes no signs or decimal points, is to be represented externally as hexadecimal; a binary edit flag of 1 indicates that the field is to be represented externally as decimal. In either case, DATENTRY ensures that only valid decimal or hexadecimal values are entered.

Date Stamp -- The date stamp code, in the form MMDDYY, indicates whether or not the field represents a date in which MM is a two-digit month number, DD is a two-digit day number, and YY is a two-digit year number. A date stamp code of 0 indicates the field does not represent a date, while a code of 1 indicates it does. Date stamp can be used only if the field has an external length of six. If the field is a zoned or character field, the internal length must also be six. If the field is packed, it must have an internal length of four. Binary fields cannot be used as date stamp fields. If the field does represent a date, DATENTRY verifies that all characters are digits, the month is in the range 01 to 12, and the day in the range 01-31. The space is filled initially with the system date, and it does not reset or appear blank after each entry.

Cumulative Field -- A cumulative field is a numeric field that accumulates the entries made through DATENTRY into one or more additional numeric fields (source fields) within a given record. The cumulative field must be defined prior to the definition of any of its source fields. Data entries made to a source field through DATENTRY are accumulated (summed) in the cumulative field. Modifications made to one or more source fields in a record cause the accumulator value for that record to be initialized and recalculated.

For example, a baseball application might consist of tallying hits comprised of singles, doubles, triples, and home runs. A HITS field first is defined as the accumulator field. Subsequently, data field definitions are specified for SINGLES, DOUBLES, TRIPLES, and HOMERUNS. Data then is entered into the source fields (e.g., SINGLES, DOUBLES), and is accumulated in the accumulator field (HITS). When the data file is reported by the REPORT utility, the accumulated values of the source fields are displayed and/or printed for the HITS field.

Field Alias -- The Field Alias option permits the operator to define an alternate name for the specified field that is to be used in the INQUIRY utility. A field alias may be up to 31 characters long, and may be composed of several words separated by single spaces.

2.3.3 Validation Specifications

Any field updatable by DATENTRY can be checked to see if the data entered conforms to user-specified criteria. Validation specifications are defined during the creation of a control file and may include the validity checks listed below.

- User Exit
- Date Stamp
- Table Look-Up
- Range

> **NOTE**
>
> Only one of the following may be specified for a given field: range, table look-up, date stamp, cumulative field, or cumulative source field.

User Exit and Date Stamp

User Exit is a data file definition option. (Refer to Sections 2.3.1 and 2.5 for details.) Date Stamp is a data field definition option, and may be used to validate date entries for month and day. (Refer to the Date Stamp description in Section 2.3.2.) Table Look-Up and Range are data field validation checks, and appear on the Field Input Validation Specifications screen following the completed definition of the data field on the Data Field Definitions display.

The Field Input Validation Specifications screen allows the user to specify the table or range checks to be made during data entry. Also displayed are the field's name and its update sequence. The update sequence is created by a counter that increases by one every time a modifiable field is defined, and presents the order in which DATENTRY displays and prompts for field values. (Refer to Chapter 3, DATENTRY, for details.) The update sequence can be modified, if desired, when the control file definition is completed. Table Look-Up and Range are described in detail below.

Table Look-Up

For checking data entered through DATENTRY, the Table Look-Up option allows the user to create and use an external validation table of legal values for the specified field. Table name is the name of the external table listing the allowable values for the field; table name may be up to six characters in length. The table name may identify a table that already exists (one table may be used by more than one field and/or data file) or one that is to be created. The table is an indexed file consisting of fixed length records.

> **NOTE**
>
> The Table Look-Up checks can only operate on fields of 16 bytes or less in length.

If the table does not exist at the time its name is given, CONTROL prompts the user for the values to be placed in the table on the table specifications screen. A number of pseudoblanks are displayed, corresponding to the number of field positions in the data field, and the user can enter the table values. Upon completion of table input, the user can proceed to create specifications for other fields. If the user chooses to enter the table values later, the table can be left blank and the values filled in later through the CONTROL Maintain Table Entries function discussed in Section 2.4.5.

NOTE

Table files created by CONTROL cannot be used by RPG II programs. To use this data with RPG II, the user must create a new, consecutive file through the CREATE user aid, and drop either the first record (for numeric tables) or the last record (for character tables).

Range

The Range Validation option allows a field's values to be checked to determine whether they are within a continuous range of values. Range validation is performed by DATENTRY when data input to a field is attempted. Low range and high range displays on the screen are used to indicate the extreme values for the field. DATENTRY will examine both range values to determine if the entered data is within the range specified. Both endpoints must be supplied by the user.

NOTE

Range validation checks can only operate on fields of 16 bytes or less in length.

2.4 MAINTAINING A CONTROL FILE

Existing control files and their table files can be maintained, modified, and examined with some of the functions listed on the Control File Utility menu. These maintenance functions are discussed in the following subsections.

2.4.1 Adding Records to a Control File

New field descriptor records can be added to an existing control file at any time by pressing PF3 from the Control File Utility menu. Therefore, the user can define additional fields for an existing control file without recreating the file. The methods used for addition of field descriptor records are the same as for creation of field descriptor records. (Refer to Section 2.3.)

Note that if fields are added to or deleted from a control file, a new data file may have to be created. Also, fields are always displayed by

DATENTRY in the order they were specified in the control file, regardless of their byte position in the actual record. Therefore, all added field descriptor records will appear after previously defined fields in the data entry screen, unless modified through the Modify Field Update Sequence screen.

2.4.2 Modifying Control File Header or Field Records

File and field descriptor records may be modified, without necessitating redefinition of the control file, by pressing PF4 from the Control File Utility menu. This feature allows the user to respecify header and field information (internal length, external length, validation criteria, etc.), replacing the old record in the control file. Note that if a field length is changed, a new data file may have to be created.

2.4.3 Deleting Records from a Control File

Field descriptor records may be deleted from the control file, without necessitating re-creation of the control file, by pressing PF5 from the Control File Utility menu. This option allows the user to retain a field only as long as it is useful.

2.4.4 Listing Records on a Control File

This option, selected from the Control File Utility menu by pressing PF6, allows the user to list the control file header and field specifications on either the screen or the printer by specifying the name of the control file. Header information is given first, including record type, record size, key and alternate key names, operations permitted, and descriptions of the file. The subsequent screen displays all specified field parameters for all fields within the control file in a tabular format.

2.4.5 Maintaining Table Entries

Table files specified for a control file can be modified or listed, or new table files can be created with options from the Table Maintenance menu. These options, which are summarized below, become available when PF7 is pressed from the Control File Utility menu. The Table Maintenance options allow the operator to enter table values at times other than when the control file is created, as well as to add, delete, and list values in a table file.

PF Key	Action
2	Create a Table File
3	Add Values to a Table File
5	Delete Values from a Table File
6	List a Table File
16	Exit to Control File Utility Menu

2.4.6 Modifying Field Update Sequence

The user may selectively respecify the sequence of defined data fields for the data entry screen when PF12 is selected from the Control File Utility menu. Defined fields are listed in their current order, and the user can modify their sequence numbers as desired. The sequence numbers specified need not be unique; duplicate numbers are processed in the order specified. When ENTER is pressed, the fields are redisplayed in their new sequence.

2.5 USER EXIT SUBROUTINES

Additional validation criteria can be enforced on the data entry process through a User Exit subroutine. A User Exit subroutine also allows the user to manipulate, perform operations on, or alter entered data as desired. Data is operated on after it is entered, but before it is transferred to the data file. Thus, a User Exit subroutine extends the user's ability to oversee the data entry process.

To use the User Exit function, the user writes a subroutine in COBOL, BASIC, or Assembler language. The user then gives it a subroutine name between USER1 and USER10 that corresponds to the User Exit subroutine name specified on the Data File Parameters display. The subroutine identifies and tests the field(s), passes a return code of 0 or 1, and specifies a user message to be returned if the validation test(s) fails. The subroutine must be linked to the DATENTRY utility through the LINKER utility, and the resulting data entry program file is named by the user. (Consult the VS Program Development Tools for details.)

The linked data entry program performs the same operations as DATENTRY, with the addition of the operations or validation checks in the User Exit subroutine. An input record is first validated according to the CONTROL specifications, following which the data is passed to the designated User Exit subroutine. The record is then tested and/or altered as specified by the subroutine. If the subroutine returns a return code of 1 to DATENTRY, DATENTRY displays the user message on the data entry screen and the record is not entered into the data file. A subroutine return code of 0 causes DATENTRY to place the data into the data file.

Section 2.5.1, Control File Requirements, describes the necessary control file specifications for User Exit processing. Construction of the User Exit subroutine is described in Section 2.5.2, Creating the User Exit Subroutine. Section 2.5.3, Linking DATENTRY to the User Exit Subroutine, treats the construction of the data entry program containing the User Exit validations and manipulations. The operation of the resulting program is discussed in Section 2.5.4.

2.5.1 Control File Requirements

The new data entry program containing the User Exit subroutine requires a control file. This control file is defined exactly as if the data were to be entered through DATENTRY, except that the User Exit option on the Data File Parameters display must be given the name of the User Exit subroutine. The subroutine name can only have the values USER1 through USER10. A control file constructed for a data entry program containing a User Exit subroutine cannot be accessed by DATENTRY; a separate control file that does not specify the User Exit option must be created for standard DATENTRY operation.

2.5.2 Creating the User Exit Subroutine

User Exit subroutines can be written in the VS COBOL, BASIC, FORTRAN, and Assembler languages. The program must conform to the specific language requirements for an external subroutine as described in the VS COBOL Reference, VS BASIC Language Reference, VS FORTRAN Reference, or VS Assembler Language Reference. The User Exit subroutine must meet additional requirements due to its link to DATENTRY. The subroutine name must be given the same name specified in the control file header record and must have a value between USER1 and USER10. For example, a BASIC subroutine contains a "SUB USER5" statement, a COBOL subroutine contains a "PROGRAM ID. USER5." statement, a FORTRAN program specifies "SUBROUTINE USER5", and an Assembler subroutine specifies "USER5 CODE".

The User Exit subroutine must return four arguments to DATENTRY in the correct order. These arguments may have any data name within the subroutine, providing that they are returned in the expected order. The arguments expected by DATENTRY are listed as follows in their return sequence.

Argument	Maximum Length	Description
Return code	1 byte	A value of 0 or 1. The subroutine must pass a return code value of 0 for successful validation and a value of 1 for unsuccessful validation.
Field name	8 bytes	The name of the field that should blink on the data entry display when DATENTRY receives a return code of 1. DATENTRY cannot signal the erroneous field unless the subroutine field name value is identical to the control file field name. Although the User Exit subroutine can manipulate more than one field, only one field can be set to blink by DATENTRY.
User message	64 bytes	A message to be displayed to the data entry operator when the record fails the User Exit subroutine validation criteria.
Record area	Record length defined in CONTROL	The record passed to the subroutine for for validation. In COBOL, fields can be defined within the record to correspond to the control file fields. BASIC and Assembler subroutines must rely on program logic to separate the desired field from the record passed by DATENTRY. The User Exit subroutine can alter the record by modifying this argument.

The subroutine logic performs the tests or operations on the field(s), assigns the appropriate return code and user message, and returns control to the main program. A sample COBOL User Exit subroutine follows. The subroutine tests the value of the first character of the EMPNO field. If the value equals zero, the program sets the return code to 1, and passes the field name and user message to DATENTRY. DATENTRY then displays the specified message FIRST CHARACTER MUST NOT BE ZERO to the user, blinks the EMPNO field, and does not accept the entered value. If the first character does not equal zero, the User Exit subroutine sets the return code value to zero, and the data is accepted.

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  USER1.
ENVIRONMENT DIVISION.
DATA DIVISION.
LINKAGE SECTION.
01  RET-URN-CODE              PIC 9.
01  FIELD-NAME                PIC X(8).
01  USER-MESSAGE              PIC X(64).
01  RECORD-AREA.
    03  FILLER                PIC X(4).
    03  EMPNO.
        05  FIRSTC            PIC X.
        05  FILLER            PIC X(4).
    03  FILLER                PIC X(71).
PROCEDURE DIVISION USING         RET-URN-CODE
                                 FIELD-NAME
                                 USER-MESSAGE
                                 RECORD-AREA.
BEGIN-IT SECTION.
VALIDATE-IT.
    IF FIRSTC IS EQUAL TO "0"
        MOVE 1 TO RET-URN-CODE
        MOVE "EMPNO" TO FIELD-NAME
        MOVE "FIRST CHARACTER MUST NOT BE ZERO" TO USER-MESSAGE
    ELSE MOVE 0 TO RET-URN-CODE.
    EXIT PROGRAM.
```

2.5.3 Linking DATENTRY to the User Exit Subroutine

In order for data to be validated by a User Exit subroutine, a new version of the data entry object program, which includes the subroutine, must be created. The creation of a new program is accomplished through the LINKER by linking the system DATENTRY program with the User Exit subroutine object code. This procedure creates a new object program file with a new name. (Refer to the VS Program Development Tools.) This linked program is then run by the user through the Command Processor or a procedure. This process may be diagrammed as follows.

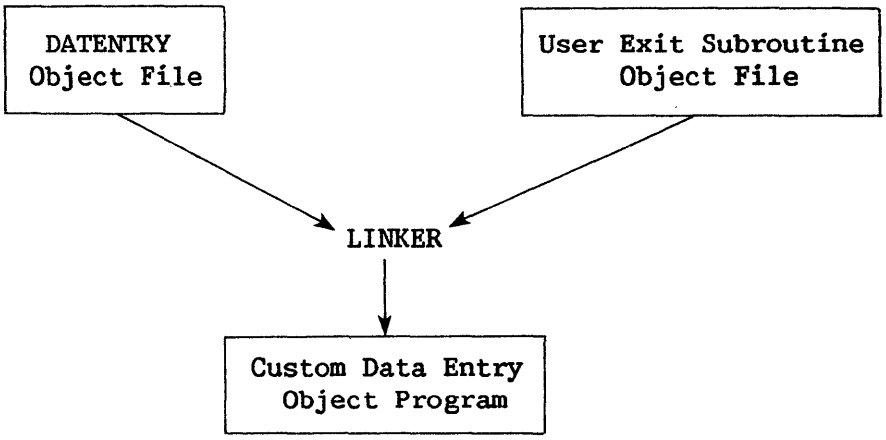


Figure 2-2. Customizing a Data Entry Program

To link DATENTRY with the User Exit subroutine, first execute the LINKER utility. Since there are no subroutines within DATENTRY, the first LINKER option screen should not be modified unless the User Exit subroutine contains additional subroutines. The object files to be linked are entered as input files in the order of their execution, and then an output file is specified. Thus, the DATENTRY utility must be specified first. Note that the output file is a data entry object program. The link is successful if a system return code of 4 or less is returned.

2.5.4 Running the Data Entry Program

Data file creation may be accomplished by executing the new data entry program through the RUN option of the Command Processor. The data entry program containing the User Exit subroutine processes information in exactly the same manner as DATENTRY, with the addition of the validation subroutine functions. The processing of data is illustrated in Figure 2-3.

NOTE

The linked data entry program must be used for data entry on all data files for which the User Exit option has been specified in the control file.

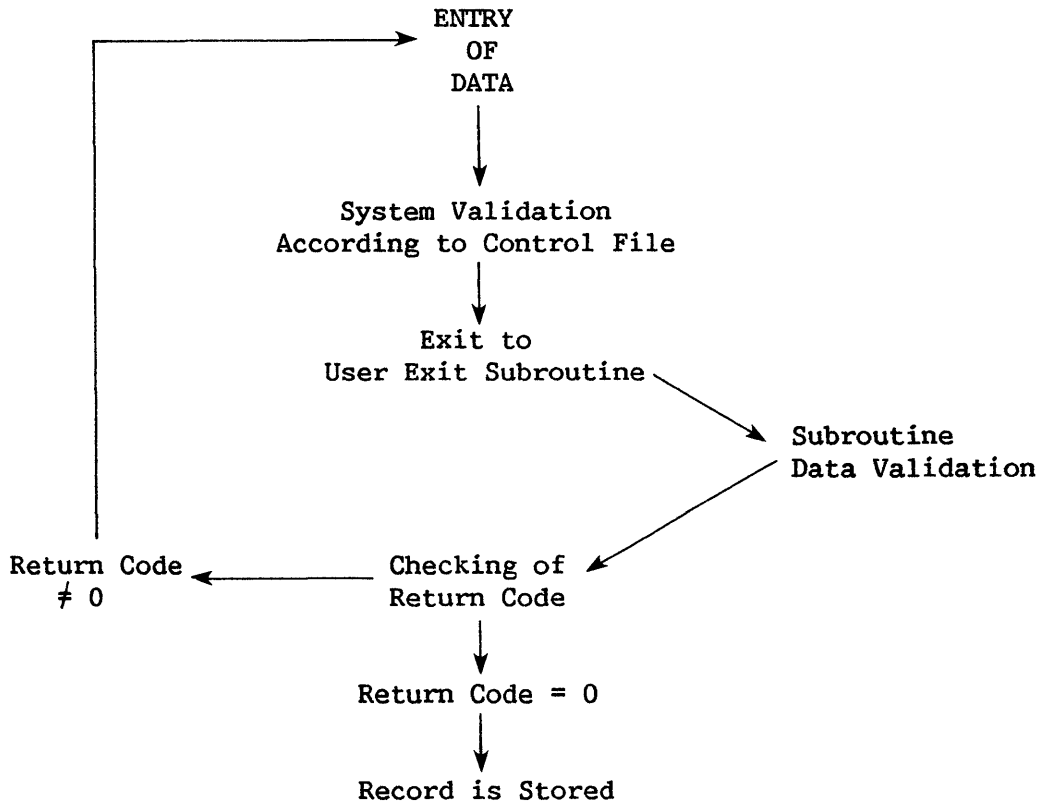


Figure 2-3. Running a Customized Data Entry Program

2.6 CREATING A SOURCE FILE FROM A CONTROL FILE

This function enables the user to create a source file from the control file. The generated source code is then inserted into the main source program by means of either the XCOPY function of the EDITOR (refer to the VS Program Development Tools) or by means of a COPY statement in the main source program. Two languages are offered: COBOL and RPG II.

The COBOL option generates 01 level record descriptions. For example, if the user wishes to insert a list of the control file fields or data items into a COBOL program, the user should first select PF8 from the Control File Utility menu and then select a name and location for the new file (which is a copy of the control file).

The RPG II option generates a source file containing file specifications, alternate index specifications, and extension specifications for table data. As with COBOL, generated RPG II source code is stored and copied into an RPG II program.

2.7 ACCESSING OTHER UTILITIES FROM CONTROL

Three related utilities, DATENTRY, REPORT, and EZFORMAT, are directly accessible from the Control File Utility menu; their associated PF keys and the chapter in which each utility is documented are summarized as follows.

PF Key	Option	Chapter
9	Run Wang VS Data Entry Utility	3
10	Run Wang VS Report Utility	5
11	Run Wang VS Screen Formatting Utility	4

2.8 A SAMPLE CONTROL PROCEDURE

A significant portion of CONTROL processing can be automated through the VS Procedure language. File locations and particular CONTROL functions can be selected with partial or no workstation interaction. However, with the exception of the Delete and Copylib functions, the procedure can only select the desired CONTROL function; further processing, such as field specification or modification, must be accomplished interactively. If the user runs the DATENTRY, REPORT, or EZFORMAT utilities through CONTROL, the procedure cannot pass parameters to GETPARMs in the called utilities. A complete list of CONTROL GETPARMs is given in Appendix A, File Management Utility GETPARMs; refer to the VS Procedure Language Reference for details concerning procedure syntax. Following is a sample procedure that runs the CONTROL utility; specifies the file and volume names of the control file (CONTROL provides a default library name); selects the Create Control File function; supplies information to the control file header record; runs the DATENTRY utility following field specification; and exits the CONTROL utility when DATENTRY processing is complete.

```
PROCEDURE
RUN CONTROL
ENTER CTLFIL FILE=CONTROL, VOLUME=SYSTEM
ENTER OPTIONS 2
ENTER HEADER RECLEN=25, FILETYPE=V
ENTER OPTIONS 9
ENTER OPTIONS 16
ENTER CTLFIL 16
RETURN
```

CHAPTER 3
DATENTRY

3.1 INTRODUCTION

The DATENTRY utility enables the user to create and maintain one or more data files described by one or more control files. Through DATENTRY, the user can add, delete, modify, or examine data in those data files.

When DATENTRY is run, the first display requests the user to supply the names of the data and control files and press ENTER. Note that when DATENTRY is called from CONTROL, the last control file name entered, as well as its library and volume names, appear as defaults.

An overview of DATENTRY processing is provided in Figure 3-1.

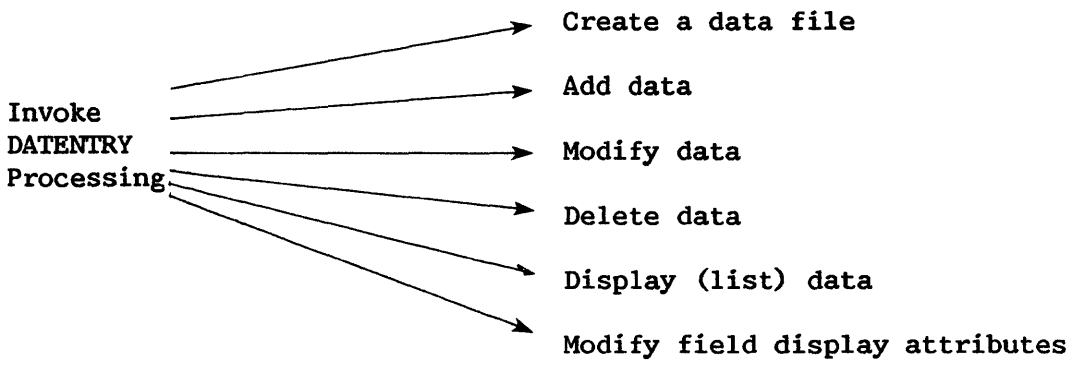


Figure 3-1. DATENTRY Processing

3.2 DATA FILE PROCESSING

Once the user has specified the names of the data and control files to be used, a screen is displayed providing the user with options to create, modify, or display a data file. The user, therefore, is able to enter, update, delete, or examine data. In addition, the user can modify certain attributes of the default data entry screen. These options, as well as the sections in which they are discussed, are listed as follows.

PF Key	Option	Section
2	Create a New Data File	3.2.1
3	Add Records to a Data File	3.2.2
4	Modify Records of a Data File	3.2.3
5	Delete Records from a Data File	3.2.4
6	List Records on a Data File	3.2.5
7	Modify Field Display Attributes	3.2.6
16	Exit to Main Screen	

3.2.1 Creating a Data File

The Data File Creation option allows the user to create a data file based on definitions that were set up in the designated control file(s). Before entering actual data, the user must supply the approximate number of records for the data file (the default is 512) to enable the system to make an initial allocation of disk space. The user may also specify the file class, retention period, whether allocated but unused disk space should be released, and storage device for the data file.

When the data file information has been specified, a second screen is displayed into which actual data records are entered. This screen presents the name of each modifiable field from the control file. Each field name is followed by a number of pseudoblanks corresponding to the external length of the particular field; data is keyed into these pseudoblanks. When ENTER is pressed, the information in each field is validated by the control file validation criteria. (Described in Chapter 2, CONTROL.) If all field entries are valid, they are entered into the data file as one record. If the value in any field is invalid, that field is highlighted and no record is written to the data file until the user corrects the value. Records containing duplicate values for the primary key of an indexed file cannot be entered. Similarly, records containing duplicate values for alternate keys not accepting duplicate values cannot be entered.

3.2.2 Adding Records to a Data File

The Data Record Addition option allows records to be added to an existing data file in the same way data is entered when first creating a data file. The data entry screen is displayed, showing field names and pseudoblanks. The user enters data directly into the pseudoblanks. An additional option is provided that allows the user to directly modify existing data records. This option is useful when a newly entered record contains erroneous data.

3.2.3 Modifying a Data File

The Data Record Modification option allows the user to retrieve, examine, and modify existing records. Options exist to position the data file at the following records.

- First Record
- Next Record
- Record (of an indexed file) whose key field or alternate key is a fully specified name
- Record (of an indexed file) whose key field or alternate key is a partially specified name (only the beginning characters of the key field are specified)
- Relative sequence number of a record in a consecutive file

The modification option also allows the user to switch directly to the Data Record Addition option. This option is useful when the selected record does not exist and the user wishes to add the record.

> **NOTE**
>
> Variable length or compressed consecutive files, primary key fields, and non-modifiable fields cannot be modified.

3.2.4 Deleting Records From a Data File

The Data Record Deletion option allows the user to delete a record of an indexed file after retrieving that record by its key field value. Records in consecutive files cannot be deleted.

3.2.5 Listing the Contents of a Data File

The option to list the contents of a data file allows the user to link directly to the DISPLAY utility where the data records may be either printed or displayed in HEX or ASCII mode.

3.2.6 Modifying Field Display Attributes

The Modify Field Display Attributes option enables the user to modify certain attributes of the default data entry screen. This option presents the user with two screens of attributes to be specified. The first screen enables the user to specify single or double spacing for all fields on the data entry screen. Double spacing allows the fields to be presented more distinctly on the screen, but accommodates only 11 lines of data. Single spacing compresses the presentation of the fields, but accommodates 21 lines of data. To select single spacing, specify SPACING=1; for double spacing, specify SPACING=2. The default (when possible) is double spacing. Pressing ENTER from the Spacing screen displays the Field Attributes screen.

The Field Attributes screen enables the user to specify the display characteristics for individual fields. The name of the field for which the attributes are being specified is displayed on the screen. Three display attributes can be specified: ULINE, DISPLAY, and UPLOW. ULINE specifies whether or not a field's entry will be underlined. The default for ULINE is NO. DISPLAY specifies how a field's value will be presented on the data entry screen. The options for DISPLAY are BRIGHT, DIM, BLINK, and BLANK; the default is BRIGHT. UPLOW determines whether or not both uppercase and lowercase alphabetic characters will be allowed for a character field's data values. The default is NO: only uppercase values are allowed if the default is retained. Use of these different field display options enables the user to highlight specific fields for data entry. For example, to call attention to a primary key field, a user might choose to have the field both underlined and blinking.

Once the field attributes for a given field have been specified, press ENTER to record these attributes and to display the attributes for the next data field. In this manner, the user can cycle through the attributes for all fields in the data file by pressing ENTER. Alternatively, the user can display the attributes for a specific field by pressing PF2 for the first field, PF4 for the previous field, or PF5 for the next field. In addition, the user can enter a field's name and press PF8 to view that particular field's attributes. The assigned field attribute information is stored in the control file, and remains as specified for subsequent user sessions unless modified. To return to the Spacing screen, press PF1; PF16 ends field display attribute specification and returns the user to the DATENTRY Options screen.

3.3 A SAMPLE DATENTRY PROCEDURE

DATENTRY is particularly well suited to execution from a procedure. GETPARMs are used for all but the actual data entry and modification screens, permitting the procedure writer to control user interaction as closely as desired. In the case of least user interaction with DATENTRY, the only screen the user ever need see is the one into which data is entered or modified.

A sample procedure that runs DATENTRY follows. This procedure displays the names of the data and control files, enters the user into the data modification option, lists the data entered, and terminates the DATENTRY program.

```
PROCEDURE
RUN DATENTRY
DISPLAY INPUT FILE=TEST, LIBRARY=IDDATA, VOLUME=SYSTEM,
        CTLFILE=TEST, CTLVOL=SYSTEM
ENTER OPTIONS 2
ENTER OPTIONS 6
ENTER OPTIONS 16
ENTER INPUT 16
RETURN
```

A complete list of all DATENTRY GETPARMs is supplied in Appendix A, File Management Utility GETPARMs. For further information concerning Procedure language, refer to the VS Procedure Language Reference.

CHAPTER 4
EZFORMAT

4.1 INTRODUCTION

The EZFORMAT utility provides three functions that facilitate interactive program development. First, EZFORMAT enables the user to dynamically design a workstation screen and to automatically generate Assembler, BASIC, COBOL, or RPG II source code that reproduces the screen format. Both messages and input requests can be specified on the screen. The code generated by EZFORMAT can be copied into a user program through the EDITOR during program development. Second, EZFORMAT can create a complete menu program in Assembler language that links user-specified programs or functions to PF keys; the user can design the workstation screen format that describes the menu. These first two functions of EZFORMAT are not related to the File Management utilities and, thus, are treated in detail in the VS System Utilities Reference.

The third EZFORMAT function, referred to as Data Entry, creates a COBOL data entry program from a user-designed screen format and a control file corresponding to the data to be entered; this function is documented in this Chapter. While the DATENTRY utility provides the ability to enter the data corresponding to a control file, the user cannot alter the appearance of the DATENTRY prompts or supply descriptive text on the data entry screen. Through EZFORMAT, the requests for input can be placed where desired on the screen, along with any explanatory text the user specifies. However, the data entry program created by EZFORMAT can only add to or modify an existing data file. Thus, the data file must be created by DATENTRY before the EZFORMAT-created data entry program can operate. Values entered through the EZFORMAT-generated data entry program can be validated by range, but table validation is not supported. An EZFORMAT data entry program cannot be generated for any data file that has nonupdatable fields defined in its associated control file; only DATENTRY can manipulate such a file.

Constructing and operating a data entry program involves the following steps, documented in the indicated sections.

- The data entry screen must be defined, as described in Section 4.2.

- The data entry program must be constructed by specifying the relationships between the screen input requests and the control file. Refer to Section 4.3 for details.

- The amount of user interaction in creating a data entry program through EZFORMAT can be reduced through the VS Procedure language. Refer to Section 4.4 for details.

- The operation of the resulting data entry program is documented in Section 4.5.

An overview of EZFORMAT Data Entry option processing is provided in Figure 4-1.

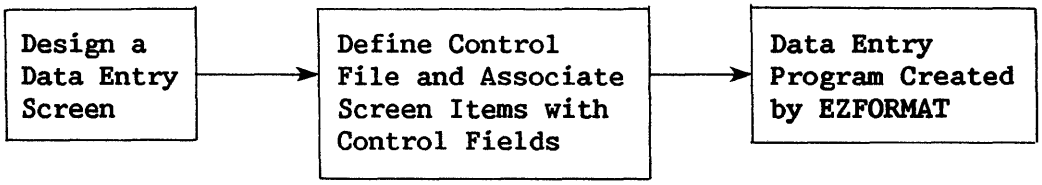


Figure 4-1. EZFORMAT Data Entry Option Processing

The EZFORMAT utility can be accessed either through the RUN option of the Command Processor or from PF11 of the Control File Utility menu of CONTROL. EZFORMAT displays the Function Selection screen when processing begins. The user selects the Data Entry option by typing "DATA ENTRY" or "D" in the pseudoblanks corresponding to OPTION on the Function Selection screen.

4.2 DEFINING THE DATA ENTRY SCREEN

The data entry screen must be defined as the first step in the creation of a data entry program. Through EZFORMAT, the user can either create a new screen format or modify an existing screen format. The Function Selection display presents the following options in addition to the selection of the data entry function.

PF Key	Function
2	Create New Format Definition
3	Use Previously Saved Screen Contents as Input for Format Definition
16	Exit without Generating Format Definition

Creation of a new format definition is documented in Section 4.2.1; Section 4.2.2 describes the process of altering a previously-defined screen format.

4.2.1 Creating a Data Entry Screen

This EZFORMAT option enables the user to dynamically arrange input data fields and message text where desired on the screen, aided by EZFORMAT functions that center or move the screen contents on request. When the Create New Format Definition option is selected from the Function Selection screen, the user is presented with the Screen Format Options display, which is an instructional screen describing the types of fields that can be specified in the workstation screen format. Three types of fields are available for screen formatting: text fields, numeric modifiable fields, and alphanumeric modifiable fields. The data entry screen is defined on a later EZFORMAT display, referred to as the Screen Definition display, which initially displays only pseudoblanks. The screen is defined by typing the text field, numeric modifiable field, and/or the alphanumeric field specifications in the desired screen locations. The differences in function and specification method for the field types follow.

Text Field — Displays nonmodifiable text at the indicated position on the workstation screen. No input is accepted in a text field when the screen is displayed in the data entry program. Text specified on the screen is considered a text field if it is enclosed in double quotation marks; the quotation marks occupy a screen position but do not display on the generated screen.

Numeric Modifiable Field — Accepts only numeric input when the user-designed screen is presented in the data entry program. The field type is identified by the specification of a 9, +, −, or any numeric character as the initial character in the field. The field is ended by a space, the end of the line, or the double quote indicator of a text field. An initial + sign generates a field that converts any entered negative data to positive values. Thus, if a −44 is entered into a field specified as +99, ½44 is stored when the data entry program is run. Negative, positive, or unsigned values can be entered into a field with an initial specification of −.

If the field specification consists of an initial 9, or a + or − followed by a 9, the data entry program displays pseudoblanks in the resulting field. If the field is specified with an initial numeric character other than 9, or if the initial + or − is followed by a number other than 9, the specified number is displayed as a default value during data entry. The initial + or − displays as a pseudoblank regardless of whether it is followed by a default value or pseudoblanks.

Alphanumeric Modifiable Field

Accepts alphanumeric input when the user-designed screen is presented in the generated data entry program. The field type is identified by an initial character other than a number, +, or -. The field is ended by a space, the end of the line, or the double quote indicator of a text field. An initial * determines that the field cannot be left blank during data entry.

If the field specification contains an initial X, or an * followed by an X, the data entry program displays pseudoblanks in the resulting field. If the field is specified with an initial character other than X, or if the * is followed by a letter other than X, the specified character string is displayed as a default value during data entry. The initial * always displays as a pseudoblank.

If ENTER is pressed from the Screen Format Options display, the Screen Format Manipulation Options menu is displayed. The Screen Format Manipulation options enable the user to move fields on the screen during the screen definition process. Rows 1 through 22 of the workstation can be formatted by the user; Rows 23 and 24 are reserved by EZFORMAT for message display during data entry program execution. Because each row on the screen begins with a non-displayed field attribute character to designate the row's default field type, only 79 columns of the workstation screen are available for user design. The screen definition process begins if ENTER is pressed from the Screen Format Manipulation Options display. The screen manipulating and field type instructional screens can be recalled during screen design: pressing ENTER retrieves the formatting options summary, while pressing PF1 displays the Field Type Instruction screen. In addition, a printed copy of the instructions can be created by pressing PF13 from the Screen Manipulation Options display. Thus, the user can refer to any needed information and return to the same stage of screen design.

The Cursor Control, NEW LINE, HOME, TAB, BACK TAB, BACK SPACE, INSERT, and DELETE keys all function to facilitate screen design on the 22 rows and 79 columns of pseudoblanks on the Screen Definition display. However, the EZFORMAT Screen Manipulation options provide more flexible functions for screen design. The functions referenced by ENTER, PF1 through PF9, PF12, and PF14 operate directly from the screen design display; PF1 and PF12 through PF16 function directly from the Screen Manipulation Options menu. The available functions are described as follows.

PF Key	Function	Description
ENTER	Display Formatting Information	Displays the Screen Manipulation Options display. The available EZFORMAT screen manipulation functions are described on the resulting display.
1	Display Field Type Information	Displays the Screen Format Options display. The available field types are described on the resulting display.
2	COLUMN	Displays the current column position of the cursor in the upper right-hand portion of the screen, temporarily overwriting any field defined in that location.
3	MOVE UP	Moves the entire contents of the row indicated by the cursor (target row) to equivalent positions in the previous row. The contents of the previous row are destroyed. The target row is replaced by pseudoblanks. No action occurs if an attempt is made to move Row 1. The function is illustrated as follows; Row 6 is the target row in the example.

Row Number	Before	After
5	AAA	BBB
6	BBB	pseudoblanks
7	CCC	CCC
8	DDD	DDD

PF Key	Function	Description
4	MOVE DOWN	Moves the entire contents of the row indicated by the cursor (target row) to equivalent positions in the following row. Any fields defined in the following row are destroyed. The target row is replaced by pseudoblanks. No action occurs if an attempt is made to move Row 22.
5	COPY UP	Copies the contents of the row indicated by the cursor (target row) to equivalent positions in the previous row. The contents of the target row are unchanged; the contents of the previous row are destroyed. Thus, the target row and the previous row are identical. No action occurs if an attempt is made to copy Row 1. The function is illustrated as follows; Row 6 is the target row in the example.

Row Number	Before	After
5	AAA	BBB
6	BBB	BBB
7	CCC	CCC
8	DDD	DDD

PF Key	Function	Description
6	COPY DOWN	Copies the contents of the row indicated by the cursor (target row) to equivalent positions in the following row. The contents of the target row are unchanged; the contents of the following row are destroyed. Thus, the target row and following row are identical. No action occurs if an attempt is made to copy Row 22.
7	ROLL UP	Moves the contents of each row, beginning with the row below that indicated by the cursor (target row) and ending with Row 22, to the row above (roll up). The contents of the target row are destroyed; the resulting Row 22 is filled with pseudoblanks. Thus, if the cursor is located at Row 19, Row 20 is moved into Row 19 (destroying the contents of Row 19), Row 21 is moved into Row 20, Row 22 is moved into Row 21, and Row 22 becomes empty. No action occurs if a roll up is attempted with Row 22 as the target row. The roll-up function is illustrated as follows; Row 6 is the target row in the example.

Row Number	Before	After
5	AAA	AAA
6	BBB	CCC
7	CCC	DDD
8	DDD	EEE
.	.	.
.	.	.
21	.	RRR
22	RRR	pseudoblanks

8	ROLL DOWN	Moves the contents of each row, beginning with the row indicated by the cursor (target row) and ending with Row 22, to the row below (roll down). The contents of Row 22 are destroyed; the resulting target row is filled with pseudoblanks. Thus, if the cursor is located at Row 19, Row 19 is moved into Row 20, Row 20 is moved into Row 21, Row 21 is moved into Row 22, and Row 22 is lost. The contents of Row 22 are destroyed if a roll down is attempted with Row 22 as the target row.
9	CENTER	Horizontally centers the contents of the row indicated by the cursor (target row).

PF KEY	Function	Description
12	TABS AND UPLOW	Displays a screen that enables the user to set up to ten tabs and to allow upper and lower case input for screen design. The tabs are set by typing a nonblank character underneath the desired column position. Note, however, that the tab positions are calculated from column one of the workstation screen, which is a nondisplayed field attribute character. Upper and lower case input is allowed if the value of MODE is changed to "UPLOW". Changing the value of MODE to UPLOW affects user input only during the screen definition process and not during the actual data entry program operation. Thus, text fields and default values for modifiable fields can have lower case values, but lower case input is not accepted in modifiable fields during data entry program execution.
13	Print Information	Creates a print file containing the information given on the Screen Format Options and Screen Manipulation Options displays.
14	Display "Finished" Screen	Displays the screen as it will appear in the data entry program. Pressing ENTER from the "final" screen returns the user to either the Screen Definition display or the Screen Manipulation Options display, depending on from which screen PF14 was pressed.
16	Exit	Ends the screen definition process.

When the data entry screen design is completed, the format is finalized when PF16 is pressed from the Screen Manipulation Options menu. The user must then select file creation options for the screen format from the File Creation Options menu. EZFORMAT can create a file containing the screen contents and/or the COBOL data entry program (generated output). The screen contents file contains only the row-by-row data for the workstation screen. The EZFORMAT generated output collectively refers to the COBOL data entry program source code, data entry program object code, and a print file containing a compiler listing of the source code and error messages. Consult Section 4.3.3 for a complete description of the generated output. Only the generated output must be saved for data entry program construction. However, if the screen contents are saved, the screen format can be modified at a later date (refer to Section 4.2.2). The following options are available from the File Creation Options menu when screen design is completed.

PF Key	Function
2	Save Generated Output Only
3	Save Screen Contents Only
4	Save Screen Contents and Generated Output
16	Exit without Saving Any Files

If the screen contents are saved, with or without the generated output, the user is immediately asked to specify the file, library, and volume names for the output file; a default library name constructed by concatenating the User ID with "SAVE" is supplied, but may be overridden. If only the screen contents are saved, EZFORMAT execution terminates once the file containing the screen contents has been named. If the generated output is saved, EZFORMAT begins constructing the data entry program as described in Section 4.3. The file, library, and volume names of the generated output are not requested until the data entry program is constructed.

4.2.2 Modifying a Data Entry Screen

This EZFORMAT option enables the user to modify the data entry screen of an existing data entry program or to modify a screen format generated for an Assembler, BASIC, COBOL, or RPG II program, providing that the screen contents have been saved as a separate file. Existing screen contents can be modified if PF3 is pressed from the Function Selection screen. EZFORMAT then requests the file, library, and volume names of the file containing the screen contents. The input library name defaults to the User ID concatenated with "SAVE", but can be overridden. The only difference in subsequent EZFORMAT processing is that the previously saved input screen contents are displayed when screen definition begins. The user can then modify the screen contents through the EZFORMAT formatting and manipulation options, save whatever files are desired, and proceed to data entry program creation.

4.3 CREATING A USER-DESIGNED DATA ENTRY PROGRAM

Once the screen format has been designed, EZFORMAT can generate a data entry program. Although EZFORMAT performs the actual code generation, the user must supply a control file and define the relationships between the fields defined in the screen format and the control file fields. Section 4.3.1 describes the specification of the control file; Section 4.3.2 describes the processes of defining relationships between the control file and screen format and specifying ranges for input values. Section 4.3.3 describes the files created by EZFORMAT when the data entry program is created. The return codes generated during data entry program creation are described in Appendix E, Return Codes.

4.3.1 Defining an Associated Control File

EZFORMAT requires a control file to initiate data entry program creation. Thus, once the File Creation options have been selected and the screen contents file, if saved, has been named, EZFORMAT requests the file, library, and volume names of a control file that corresponds to the input data fields defined on the data entry screen. If such a control file does not exist, the CONTROL utility can be invoked without interrupting EZFORMAT processing by pressing PF2 from the EZFORMAT Control File Definition screen. Following definition of the control file, DATENTRY can be invoked from CONTROL to create the data file that is to be modified by the EZFORMAT data entry program. When the control file has been defined (refer to Chapter 2, CONTROL, for details) and the data file created, if desired, EZFORMAT processing continues as if it had just received the file parameters of a previously created control file. Note that no data entry program can be generated if the control file contains nonupdatable fields. The user can also choose not to create a data entry program and either press PF1 to switch to the COBOL option of EZFORMAT or press PF16 to terminate EZFORMAT processing. If the user exits EZFORMAT, the COBOL source code file is not created regardless of which screen format file creation option was previously selected.

4.3.2 Correlating Control File Field Names with Screen Format Names

Because the input data fields on the screen format need not be placed in the order of the fields in the control file, the user must associate each input data field defined on the screen with a control file field name. EZFORMAT displays a list of control file field names immediately following the Control File Definition screen. The user can then proceed to the Field Correlation screen by pressing ENTER; the list of control file field names can be retrieved by pressing PF15 from the Field Correlation screen. The data entry screen format can be reviewed if PF14 is pressed.

The Field Correlation screen has an entry for each modifiable field defined on the screen format. The entries are listed in the order they appear on the screen, i.e., left to right, top to bottom. Field correlation is accomplished by supplying information to four types of values for each field; the types of values describing each modifiable field are described in detail as follows.

FIELD NAME — The name used by the COBOL data entry program to refer to the screen location. EZFORMAT automatically generates a name of the form ROWXX-COLYY, where XX and YY represent the 2 digit row and column positions, respectively. This value can be changed by the user to an alternate value providing it does not duplicate a source or object name or a COBOL reserved word. However, it is not necessary to change the EZFORMAT-generated field name.

SOURCE

The field from which FIELD NAME obtains its initial contents when the data entry program is run. Thus, the SOURCE name must be equal to the name of the control file field corresponding to the screen location. EZFORMAT supplies names reflecting the data type and the field occurrence; i.e., "ALF-OBJ001" reflects an alphanumeric modifiable field which is the first input request, and "NUM-OBJ005" reflects the fifth input request, which happens to be a numeric modifiable field.

NOTE

The default names are only provided to assist the user in identifying the appropriate field and must be altered to indicate the control file field name.

However, if the modifiable field has been supplied with a default value during screen definition, no SOURCE name can be assigned by either EZFORMAT or the user; the default value specified during screen definition will always be displayed by the data entry program, regardless of the actual contents of the data file.

OBJECT

The field to which the contents of FIELD NAME are transferred when a value is supplied through the data entry program. Thus, the OBJECT name must be equal to the name of the control file field corresponding to the screen location. EZFORMAT supplies a default name constructed in the same manner as SOURCE default names; however, this name must be changed to indicate the control file field name. For the typical data entry program, the same name value is specified for OBJECT as for SOURCE.

RANGE

The upper and lower bounds of values to be accepted as input by the data entry program.

NOTE

Range values specified in the CONTROL utility do not function in an EZFORMAT data entry program; thus, only those range values that are specified during EZFORMAT operate when entries are added to the data file or data file entries are modified during data entry program execution.

If an alphanumeric modifiable field was defined with an initial *, EZFORMAT supplies default upper and lower bounds to ensure that a blank value cannot be entered. The COBOL collating sequence determines the interpretation of the upper and lower bounds; consult the VS COBOL Reference for details. RANGE values are optional for data entry fields.

The user must associate a control file field, and optionally assign range values and alter the FIELD NAME, for each field defined in the screen format. When all the fields displayed on the first screen have been correlated, any subsequent fields can be displayed by pressing ENTER. The user can return to the first screen of fields by pressing PF1. When field correlation is complete, the operation is terminated by pressing PF16.

4.3.3 EZFORMAT Output Files

When field correlation is complete, EZFORMAT automatically creates the data entry program. EZFORMAT generates a COBOL source file corresponding to the data entry program; this file can be modified by the user in the EDITOR if further custom design is desired. EZFORMAT requests the file, library, and volume names of this file immediately following field correlation; a default library name constructed by concatenating the User ID with "COPY" is supplied, but may be overridden.

Once the COBOL source code for the data entry program is constructed, EZFORMAT creates the corresponding object code. EZFORMAT then requests the file, library, and volume names for the data entry program object code to be created. A print file with the same name as the COBOL source code is placed in the user print library; this file contains the compiler listing and contains a source and error listing for the data entry program. EZFORMAT processing ends, returning the user to the Command Processor, and the output files can be accessed. EZFORMAT return codes are listed in Appendix E, File Management Utility Return Codes.

4.4 A SAMPLE EZFORMAT PROCEDURE

Although screen definition in EZFORMAT is by nature an interactive process, a significant amount of user interaction in EZFORMAT processing can be eliminated. Through a procedure, the EZFORMAT function, file creation options, and output file locations can all be selected. If a data entry screen has been previously defined, all user interaction except field correlation can be eliminated. Consult the VS Procedure Language Reference for details concerning procedure syntax.

Although the Function Selection screen does not solicit information through a GETPARM, the utility contains a hidden GETPARM, identified by the OPTIONS prname that duplicates the purpose of the Function Selection screen. Thus, the Data Entry function is selected through the LANGUAGE keyword, PF2 effects the creation of a new screen format, and PF3 indicates that previously created screen contents are to be used as input to EZFORMAT. Without procedure control, the user must enter the screen definition process even if the input screen format is in final form; the hidden GETPARM additionally allows the user to bypass screen definition completely through the PROCEDUR

keyword. The value of the PROCEDUR keyword is only considered when EZFORMAT processes previously created screen contents. The default value, YES, bypasses the screen definition phase of EZFORMAT and continues EZFORMAT processing with the File Creation Options screen.

A complete listing of the EZFORMAT GETPARM requirements is given in Appendix A, File Management Utility GETPARMs. However, a sample procedure that runs the EZFORMAT utility is provided below. The procedure selects the Data Entry function using predefined screen contents, supplies the name of the input screen contents file, chooses to save only the data entry program as output, invokes the CONTROL utility, and names the COBOL source and object code of the data entry program. Notice that the procedure cannot supply values to CONTROL GETPARMs since procedures cannot pass values to called programs. Also note that the value of LANGUAGE must be specified as "D", rather than "DATA ENTRY", because the Procedure language does not accept embedded blanks.

```
PROCEDURE
RUN EZFORMAT
ENTER OPTIONS 3, LANGUAGE=D, PROCEDUR=YES
ENTER INSCREEN FILE=EZENTRY, LIBRARY=MLHSAVE, VOLUME=SYSTEM
ENTER FUNCTION 2
ENTER CONTROL 2
ENTER COPYLIBR FILE=EZENTRY, VOLUME=SYSTEM
ENTER PROGRAM FILE=EZENTRY, LIBRARY=MLHOBJ, VOLUME=SYSTEM
RETURN
```

4.5 RUNNING THE USER-DESIGNED DATA ENTRY PROGRAM

The data entry program generated by EZFORMAT enables the user to add or modify entries in an existing consecutive file and add, modify, or delete entries in an existing indexed file. If the data entry program was generated for an indexed file, processing begins with a request for the existing data file's file, library, and volume location. When the indexed file parameters have been supplied, the data entry program displays the Function Selection menu. However, if the data entry program was generated for a consecutive data file, the data entry program initially displays the Function Selection menu, and does not request the existing data file's parameters until a specific data entry program function is selected. The Function Selection menu provides the following options; however, the delete function does not appear on the menu if the control file indicates a consecutive file.

PF Key	Function
3	Add Entries to the File
4	Maintain the Entries in the File
5	Delete Entries in the File
16	Exit from the Program

Extending a data file is treated in Section 4.5.1; Section 4.5.2 discusses the process of changing the values of data file entries. Section 4.5.3 describes the process of removing records from an indexed data file.

4.5.1 Adding Entries to a Data File

When this option is selected from the data entry program Function Selection menu, the data entry screen designed in EZFORMAT is displayed. Any default values specified in EZFORMAT are displayed in their respective fields on the data entry screen. The user can then enter the data on the screen. When all required fields have been supplied, the values are added to the data file if ENTER is pressed. Alternatively, pressing PF16 returns control to the Function Selection menu without adding the current screen contents to the data file. Data is always added to the end of an existing consecutive file. Entries added to an indexed file are placed in primary key sequence. If an invalid (wrong data type or out of range) value is entered, the program redisplays the screen contents with all invalid entries blinking and the cursor located at the first invalid entry. The user can then correct the entry(s), and press ENTER again. When all fields have been correctly specified, the data entry program again displays the data entry screen designed in EZFORMAT to accept input for the next record. The Function Selection menu is retrieved when data entry is completed by pressing PF16; the user can then modify any entries or exit the program.

4.5.2 Modifying Data File Entries

Existing data file entries can be modified when PF4 is pressed from the Function Selection menu. Entries can be accessed either sequentially or randomly for both consecutive and indexed files; however, the access method differs for the two file types. The records are retrieved through the Record Selection menu, which is displayed when PF4 is pressed from the Function Selection menu. For either file type, the first record can be retrieved for modification by pressing PF2; the record following the current record can be accessed by pressing PF3. The user can thus proceed sequentially through the file, modifying the erroneous entries.

Sequential access may not be convenient for a long consecutive data file. Consecutive files can be accessed randomly by specifying the record number (i.e., 1, 33, 11, etc.) in the KEY field and pressing ENTER. Indexed files can be randomly accessed according to the primary key. For indexed files, the name of the primary key field is displayed with a prompt for a value; if the desired value is entered and ENTER is pressed, the corresponding record is obtained.

However the record was accessed, the selected record is displayed in the format of the data entry screen designed in EZFORMAT. If a default value was supplied for any field, the default value is displayed in the field, regardless of the field's current contents. The primary key field of an indexed file cannot be modified; the contents of any other field can be altered. If ENTER is pressed, the record is replaced with the altered screen contents, providing an invalid value has not been entered. If an invalid value is entered, the data entry program prompts the user in the same manner as described for adding entries to a data file. When valid values have been entered, the user is returned to the Record Selection screen to choose another record. However, the user can always return to the Record Selection screen without changing the file contents by pressing PF16 instead of ENTER. Record

modification can be terminated at any time by pressing PF16 from the Record Selection screen. The data entry program then displays the Function Selection menu from which records can be added, modified, or deleted, or data entry program processing terminated.

4.5.3 Deleting Data File Entries

Only indexed data file records can be deleted through the EZFORMAT-generated data entry program. The user is presented with a Record Selection screen from which to retrieve the record to be deleted when this option is chosen from the Function Selection menu. Sequential record access is not supported for deletion. The user must specify a primary key value and press ENTER to retrieve a record. The record is then displayed in the data entry screen format. Pressing PF3 deletes the displayed record from the file and redisplays the Record Selection screen, while pressing PF16 returns the user to the Record Selection screen without deleting the displayed record. The user can then retrieve another record for possible deletion, or return to the Function Selection menu by pressing PF16.

CHAPTER 5
REPORT

5.1 INTRODUCTION

The REPORT utility provides a method of creating, modifying, and printing reports on data files. Designed especially for nontechnical users, REPORT features a step-by-step approach that leads a user through the process of defining a report. REPORT also allows modification of the report definition. The Modify option uses, to a large extent, the same displays as the report definition phase and makes it easy for the operator to modify an existing report. In addition to producing formal reports, this utility also enables the user to obtain listings of data that fall within specified limits, for example, a listing of the salesmen selling more than $10,000 during a specific month. INQUIRY, which is discussed in Chapter 6, provides a more elaborate way of extracting data based on user-specified criteria.

The process of producing a report involves the following steps, documented in the indicated sections.

- The report format and content must be defined. For details, refer to Section 5.2.
- An existing report format and content can be modified. For details, refer to Section 5.3.
- Once the format and content have been defined, the report is printed. The steps involved in printing a report are discussed in Section 5.4.
- The report can be further customized through a User Exit subroutine, discussed in Section 5.5.
- The amount of workstation interaction in producing a report can be reduced through the VS Procedure language, as described in Section 5.6.

An overview of REPORT processing is provided in Figure 5-1.

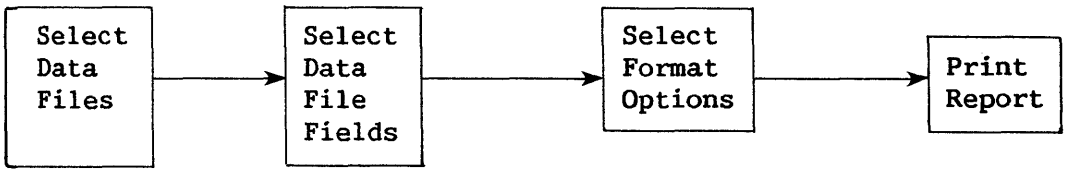


Figure 5-1. REPORT Processing

REPORT can be invoked directly from the Command Processor or from the CONTROL File Utility menu. The first REPORT screen, referred to as the REPORT Utility menu, offers the following options.

PF Key	Option
2	Create a Report Definition
3	Modify Report Attributes
4	Print a Report
16	End Job

5.2 CREATING A REPORT DEFINITION FILE

The first step in producing a report is to create a Report Definition File (RDF). Creating the RDF can be simple or complex, depending on the requirements of the user. Almost all screens have default field values to be used if the user does not wish to modify them. REPORT creates the RDF by combining the default values with the values entered by the operator. Once an RDF has been created, the report can be run repeatedly by specifying its Report ID in the utility's print option.

5.2.1 Report Definition File Creation Options

The Report Definition file can be defined interactively when PF2 from the REPORT Utility menu is pressed. The Creation Options screen is displayed when the RDF Creation option is selected. Through the Creation Options screen, the user names the RDF and selects additional REPORT functions.

The Report ID is the name used by REPORT to identify the RDF. The value specified for the Report ID is the VS file name; a default library name is constructed for the RDF by concatenating the User ID with RPT. If a default OUTVOL has been set by the user through the Command Processor or a procedure, the RDF file parameters are the Report ID, default library, and default volume names. If no default OUTVOL has been specified, the user is asked to select an output volume following the field selection phase of report processing. (Refer to Section 5.2.3.) The Report ID and default library names can be changed when the output volume is specified. Wherever possible, the Report Definition file should have the same name as its associated data and control files.

The Creation Options screen also allows the user to select one of two REPORT options. The user can choose to report on more than one data file to invoke a User Exit program for additional report format control. The default selection response for each option is NO, so that the user need not specify a response for any option unless a response is desired. If a secondary DMS file is to be used in the report, the file must be indexed because it is chained to the primary file through its primary key field. User Exit program processing is described in Section 5.6.

5.2.2 Data File Selection

The user must specify the name of the primary data file on the Primary Data File Selection screen. The primary data file on which the report is to be based is specified by providing its file name and, optionally, the library and volume names. If the library and/or volume are not specified, their names are obtained when the report is printed, from either user input or system defaults specified by the user. The primary data file is the only file on which a sort can be performed.

REPORT assumes that the data file has a corresponding control file with the same file name in the default control file library. Unless a default INVOL has been set, the user must specify the volume location of the control file and/or supply the actual file and library names of the control file if they differ from the default values. If a second data file is to be used for the report, the file, library, and volume names are requested following primary data file field selection. (Refer to Section 5.2.3, Field Selection.) On the Secondary Data File Selection screen, the user must provide the name of the field in the primary file to be used for chaining the primary file to the secondary file. This field must be the key field of the secondary file. REPORT assumes that the secondary data file has an associated control file with the same file name; the control file parameters must be specified for the secondary data file just as for the primary data file. The user can then select fields from the secondary data file, as described in Section 5.2.3.

5.2.3 Field Selection

The user must select the fields needed for report preparation from the list of all of the control file's fields presented on the Primary File Field Selection screen. Fields used merely for record selection or sorting purposes should be selected, as well as fields to be printed in the report. It may be useful to select extra fields in anticipation of future requirements. (These extra fields can be assigned a 99 sequence number and need not be used for report production. Refer to Field Sequence, in Section 5.2.4, for additional information.)

If a second file is to be used for the report, its fields are presented for user selection on the Secondary File Field Selection screen following secondary data file selection. The fields are selected by the same process as the fields in the primary file.

Following field selection for the primary and/or secondary files, the user can specify new fields to be used in the report. New fields comprise either existing numeric fields or constants whose values are added, subtracted, multiplied, or divided with other numeric fields and/or constants, or existing character fields or literals whose values are concatenated with other character fields and/or literal values. The arithmetic operations used in creating new fields are performed in the order in which they are specified. If new fields are desired, they are defined on subsequent screens by giving the field IDs, field types, lengths (i.e., the number of characters), decimal positions (if field type is numeric), source fields, and the arithmetic operations to be performed.

The sum of the primary, secondary, and new fields selected cannot exceed 80. The total width of all fields appearing on any one print line in the report cannot exceed 132 bytes; the total width for all print lines cannot exceed 396 bytes. The width of a field is determined by the width of its header or the width of its contents, whichever is greater.

5.2.4 Report Definition Options

After all fields have been selected for the report, the following 13 options become available and are displayed on the Report Definition Options menu. These options determine the report format and are described in detail in this section. Pressing ENTER from the Report Definition Options menu enables the user to proceed sequentially through each option screen, specifying the desired format for each option. Default values are provided for all options so that in many cases the user can quickly define the report format. After each screen has been filled, pressing ENTER displays the next option in sequence, through the PF11 option. The functions associated with PF keys 12 and 16 are only available from the Report Definition Options menu.

PF Key	Action	Default Action
1	Display the Report Definition Menu	None
2	Report Title Information	Report ID
3	Column Headings	Field IDs
4	Spacing Between Fields	2 spaces
5	Field Sequence on Report	Alphabetic Order
6	External Field Size	Per Control File
7	Edit Options	Per Control File
8	Data Limits for Record Selection	None
9	Sort Fields	None
10	Control Fields	None
11	Report Summary Options	None
12	Print Headings and Dummy Detail Lines	
16	Exit Report Definition Phase	

The fields selected may result in a print line exceeding 132 characters. In that event, a subset of the options is provided so that the user can reduce the number of characters in that print line. The user cannot proceed with report definition until the print line size is diminished by (a) reducing the external field size, the spacing between fields, or the length of one or more column headings; (b) redefining which fields are to be included in the report; and/or (c) excluding one or more fields from the report or respecifying on which lines the fields are to appear through the Field Sequence option.

Report Title and Headings Information

Three types of titles may be specified for a report: report title, page heading, and/or control break descriptors.

Report Title	The report title may be up to three lines long, is required, and always appears on the first page.
Page Headings	If page headings are selected, they appear on all pages except the first page. If page headings are not selected, the report title appears on all pages. The report date (selected at report production time) and page numbers appear at the top of every page.
Control Break Descriptors	A control break descriptor is a user-supplied subtitle appearing within the body of a report whenever the value of a control field changes (e.g., at each control break). Control break descriptors appear on the same line as the associated total and in the selected column. (The default is column 1.) (Refer to Control Fields and Report Summary Options, within this Section.)

Column Headings

The field ID (field name) is used as the field's column heading unless a different heading is provided. The space allocated for the field on the report is the column heading length or external field size, whichever is greater. The column heading can be blank. User-selected column subheadings may also be defined at this time.

NOTE

Lines for column headings for print lines 2 and 3 are reserved, even if no column heading for fields appearing on these lines are specified. If column headings for fields on print lines 2 and 3 are specified, these headings are printed above their corresponding fields, on their respective lines.

Spaces Before Fields

Two spaces normally are placed before each field on the report. The user may change the number of spaces preceding the first character of each field and its column heading. If the total number of spaces specified for any print line exceeds 132 characters, the Print Line Reduction screen appears.

Field Sequence

With the Field Sequence option, the user can specify both the line number (1-3) and the sequence number (1-80) for each field to be printed in the report. Fields given a sequence number of 99 do not appear on the report, but may be used in the preparation of the report. For example, if it is desirable to sort data by the field DEPT but not to print the values for DEPT, DEPT is given a sequence number of 99. The field names are presented to the user in sorted (i.e., alphabetic or numeric) order, with all fields assigned to line 1; unless modified, these values are the defaults. The user can modify both the line numbers and sequence numbers by typing over the displayed values.

When determining the sequence of a large number of fields, skipping numbers is useful (e.g., numbering by 5s). Skipping numbers allows fields to be easily inserted between two other fields in sequence. Also, by skipping numbers, more than one field may be given the same sequence number. In this case, the first field of a given sequence number is placed before subsequent fields of the same sequence number, etc. The fields then are automatically renumbered in increments of one and rearranged by sequence number within line number when ENTER is pressed. This duplicate numbering option facilitates ordering of large numbers of fields.

External Field Size

The External Field Size option is used to specify the number of character positions each field is to occupy on the report. The default size indicated is the number of character positions specified in CONTROL. Fields can be increased in size by padding with blank characters, or they can be truncated. Character fields are left-justified and are truncated from the right, while numeric fields are right-justified and are truncated from the left. Adding dollar signs, commas, or other edit characters through the data edit options increases the size of the field, and must be considered by the user to avoid unwanted truncation. (Refer to Data Edit Options, in this Section.) The external field size may not be zero.

There are three primary reasons to alter the external field sizes for a report.

Reduced Size -- to delete undesired or irrelevant data by truncation.

Increased Size -- to allow space for a field that has been expanded due to special character insertion (", ", "$", "/", "-").

Total Values -- to allow space for a numeric field's total value (created through the Report Summary options) when the total value exceeds the field's length.

Data Edit Options

The user has the option of altering the data format for any numeric field (packed, zoned, unsigned, or binary if the Binary Edit Code = 1) scheduled to appear on the report. The Data Edit Options screen presents a "picture" of each field, and requests the user to specify whether or not any given field is to be modified. If modification is chosen, an edit screen appears requesting the user to select which modifications are to be performed. The current format is displayed as the default. Edit options for numeric fields are listed as follows with explanations of the entry codes. Entry codes can also be displayed by pressing PF14.

Suppress Zeroes

ZN	Zero-suppress leading zeroes except for rightmost N positions.
*N	Asterisk-protect leading zeroes except for rightmost N positions.
Blank	No zero suppression.

Sign Control

CR	Print trailing 'CR' if the field is negative.
DB	Print trailing 'DB' if the field is negative.
9-	Print 1 trailing blank if the field is positive, trailing '-' if negative.
Blank	Print no signs.

Decimal Carry

.N	Print out N characters to the right of the decimal point.
Blank	Print all decimal positions, but no decimal point.

Special Insertion Characters

(Notes: N cannot be zero.
A leading dollar sign is not included in determining N.)

N*	Insert an '*' to the right of the Nth character (leftmost = 1).
N-	Insert a '-' to the right of the Nth character.
N,	Insert a ',' to the right of the Nth character.
N/	Insert a '/' to the right of the Nth character.
N.	Insert a '.' to the right of the Nth character.
N	Insert a ' ' (blank) to the right of the Nth character.
Blank	No special characters.

Dollar Sign

$9	Insert a "$" in the leftmost field position.
$$	Float the dollar sign.
Blank	No dollar sign.

Commas

Y	Insert a comma after every 3rd integer from the decimal point.
N	No comma insertion.

Data Limits

The Data Limits option allows the user to select records for printing by defining limits on the values of up to ten fields within the records. The legal operators are listed as follows.

GT	Greater than
GE	Greater than or equal to
LT	Less than
LE	Less than or equal to
EQ	Equal to
NE	Not equal to

Up to ten of these conditions can be specified for each field, connected with AND/OR operators. The AND operators are evaluated first. Consequently, a set of limits is defined for record selection for use in report printing. If two or more conditions are connected with AND, the record is selected only if both (or all) conditions hold true. If the OR connector is used, the record is selected if at least one of the conditions is true.

For example, suppose each record consists of the following fields: employee's name (NAME), social security number (IDNUMBER), department (DEPT), hours worked during preceding week (HOURS), and hourly wage rate (WRATE). The users desire a separate report on all employees who worked fewer than 40 hours and whose wage rates exceed $4.00. They therefore define the limit on the HOURS field as LT 40, the limit on WRATE as GT $4.00, and specify AND as the connector. Any record with values outside of these limits is not printed.

File Sort

The File Sort option allows the user to define the order of records to be printed on the report. Up to eight fields from the primary file can be sorted in either ascending or descending order.

The user assigns the highest level to the primary field on which the sort is based. The next level would be assigned to the secondary field. For example, if the file is first sorted, according to department, DEPT will be assigned Level 1. If, within each department, the file is to be sorted alphabetically according to the employees' family names, NAME will be assigned to Level 2.

Control Fields

Control fields govern the totalling and printing of subtotals at report time, and provide the facility to include breaks within the report with either spacing or page ejections. Page ejection facilitates sending selective portions of a report to different departments. The specified break occurs whenever the value of a control field changes.

In the printed report, a subtotal line for all fields selected for total values (through the Report Summary Options screen) is printed following the printed field values of the control field section. If a control field descriptor is specified, it also appears on this subtotal line. Note that careful placement of the descriptor through the Spaces Before Fields screen is necessary to avoid overprinting of descriptors and data. If no totals are specified in the Report Summary Options screen, this line displays only the descriptor, if selected, or nothing if no descriptor and no totals are specified.

Each control field must be assigned a level number. The level number is used to associate each total line (if specified through the Report Summary Options screen, or blank line if not), with a control break descriptor (if specified through the Report Title Information screen, or blank if not). Level 1 control breaks occur most frequently, and are associated with Level 1 control break descriptors.

Report Summary Options

Report Summary options are listed as follows.

TOTAL
MAXIMUM
MINIMUM
AVERAGE

The Report Summary information is provided at the end of the report for the values contained in each numeric field selected. If control fields have been defined, subtotals are provided at control breaks for all numeric fields (if any) for which totals are requested. (Refer to Control Fields in this Section.)

Print Headings and Dummy Detail Lines

The Print Headings and Dummy Detail Lines option prints out a simulated run of the report, including the title lines, heading lines and one detail line. The detail line uses Xs for character fields, 9s for numeric fields, Fs for binary hexadecimal fields, plus any editing character selected (such as $) to indicate the formats and positions of the fields.

5.3 MODIFYING A REPORT DEFINITION FILE

The Report Modification option of the REPORT utility is invoked by selecting PF3 from the REPORT Utility menu. The Report Definition file to be modified must first be identified. If the RDF does not reside in the default library and/or if the user has no default INVOL, REPORT requests the file, library, and volume names of the RDF immediately after receiving the Report ID. The control and data files to be used in the report can be respecified on a screen that appears when PF8 is pressed from the Report Modification Screen. In this way, the user can change the data and/or control files used in preparing the report. In addition, the User Exit option can be added or deleted on the same screen as control and data file respecification. If the user specifies the file, library, and volume names of a User Exit subroutine, the report format is controlled by the specified User Exit subroutine. Similarly, the User Exit option can be deleted from the RDF by erasing the values specified for file, library, and volume names.

The next screen enables additional primary data file, secondary data file, or new fields to be included in the report if the user overrides the default NO response. If new fields are defined, the user first defines the field name, data type, and length, and then defines the new field's components by choosing to modify them. All added fields are given a 99 sequence number by default; this value may be changed at a later stage of report modification. All options available on the Report Definition Options menu then become available for report modification, and are discussed in the following Sections. The only restriction on modification is that the data fields previously specified for the report may not be changed.

5.3.1 Report Title Information

Report titles, page headings, and control break descriptors may be modified by the user. Refer to Report Title and Headings Information in Section 5.2.4 for additional information.

5.3.2 Column Headings

Column headings and subheadings (maximum length 25 characters) may be modified by the user for each field. Refer to Column Headings in Section 5.2.4 for additional information.

5.3.3 Spacing Before Fields

The spacing before fields on the report may be modified by the user. For additional information, refer to Spaces before Fields in Section 5.2.4.

5.3.4 Field Sequence on Report

The user can modify the sequence and line numbers of the fields on the report. A field also can be given a sequence number of 99, which means that the field can be used in report preparation (e.g., for data limit purposes), but the field does not appear on the report. For additional information, refer to Field Sequence in Section 5.2.4.

5.3.5 External Field Size

The user can alter the external field size of any field to add blanks to the field or to truncate part of the field. For additional information, refer to External Field Size in Section 5.2.4.

5.3.6 Edit Options

The user can alter the data edit format options selected in the report definition. For additional information, refer to Data Edit Options in Section 5.2.4.

5.3.7 Data Limits for Record Selection

The user can alter or delete the limits to be imposed on a field to determine which records should be printed on the report. Up to 10 fields may be defined with limits, and up to 10 AND/OR connectors and operands may be used per field with each set of limits. For additional information, refer to Data Limits in Section 5.2.4.

5.3.8 Sort Fields

The sort fields selected at the report definition stage can be altered by the user. The records to be used in the report can be sorted on up to eight fields, with any combination of ascending or descending order on the individual fields. However, the fields to be sorted must all reside in the primary file. For additional information, refer to File Sort in Section 5.2.4.

5.3.9 Control Fields

The control fields selected during the report definition stage may be altered. Up to five levels of control fields may be selected for the report. For additional information, refer to Control Fields in Section 5.2.4.

5.3.10 Report Summaries

Summary information to be printed at the end of the report may be altered by the user. Summary information can be printed for any and all numeric fields in the report and includes totals, maximum values, minimum values, and averages. For details, refer to Report Summary Options in Section 5.2.4.

5.3.11 Print Headings and Dummy Detail Lines

This option enables the user to print the title lines, heading lines, and a detail line of the defined report. In this way, the user can review any modifications made through the Report Modification option of the REPORT utility. Refer to Section 5.2.4 for further information.

5.4 PRINTING A REPORT

When PF4 is pressed from the REPORT utility menu, the Print Report screen is displayed. From the Print Report screen the user can specify certain parameters for the printing of previously defined reports. These parameters include the report date, output device, data file changes, record count, lines per page, print lines to be printed, and the order of the print lines. If the library and volume names of the primary and/or secondary data files have not been previously specified, their values must be supplied on the screen that appears following the RDF definition. The library and volume names can also be given immediately following the Print Report screen, if no additional file parameters were required for the RDF. The Print Report options are discussed in detail in the following subsections.

5.4.1 Report ID

The Report ID field specifies the name of the report definition file to be used in printing the report. The next screen requests the file parameters of the RDF if the default values are incorrect or if a default INVOL has not been specified.

5.4.2 Report Date

The selected report date appears on the top of every page. The default for the report date is the current date.

5.4.3 Output Device

The user can choose either to display the report on the screen (DISPLAY) or to print it (PRINTER). The default is PRINTER.

5.4.4 Change Data Files

The user can elect to change the data file(s) to be used for the report. The response is YES or NO; the default is NO. The affirmative response produces a screen for data file respecification.

5.4.5 Count Option

The Count option allows the user to select a number of records to be used in the report. This option can be used either to give a count of the records processed or to limit the number of records to be listed (as in a "TOP TEN" report).

5.4.6 Sum Only

The Sum Only option provides the capability to obtain a report of total information (i.e., no detail lines). Only control break and total lines are printed. Totals appear only if they were specified in the report file specifications.

5.4.7 Lines Per Page

The Lines Per Page option permits the user to specify any number from 05 to 99 as the maximum number of lines printed per page.

5.4.8 Select Lines

The Select Lines option allows the user to specify the order of the three print lines (and their corresponding column heading lines, if any) by specifying the digits 1, 2, and 3 in the desired order. Since a user can enter a leading or trailing space in place of a number, the user also selects which print lines are printed. Two examples of the Select Lines option are as follows.

Select Lines = 123 prints as:	Select Lines = 32 prints as:
LINE 1	LINE 3
LINE 2	LINE 2
LINE 3	

5.4.9 Print Line Spacing

The Print Line Spacing option permits specification of the number of blank lines following each print line, in the order specified in the Select Lines option. Up to 5 blank lines can be specified after each print line. Print line spacing is illustrated as follows.

Select Lines = 213 Print Line Spacing = 031	Select Lines = 123 (only 1 data line specified) Print Line Spacing = 000
LINE 2	LINE 1(A)
LINE 1	LINE 1(B)
blank line	LINE 1(C)
blank line	.
blank line	.
LINE 3	
blank line	

5.5 INVOKING A USER EXIT SUBROUTINE

The report format and content defined in REPORT can be overridden by the user on a record-by-record basis through a User Exit subroutine. Through a User Exit subroutine, the user can omit a line from the report, print additional lines, or alter the contents of a line constructed by REPORT. Invoking a User Exit subroutine involves the following steps.

- The Report Definition File must indicate User Exit processing and contain the file parameters of the appropriate subroutine. Refer to Section 5.5.1 for details.
- A compiled or assembled User Exit subroutine must be constructed. Refer to Section 5.5.2 for details.
- The report is printed with the subroutine manipulating the format and content. Refer to Section 5.5.3 for details.

5.5.1 Report Definition File Requirements

For the User Exit subroutine to function, the RDF must have the User Exit option specified and must contain the file parameters of the User Exit subroutine object code. If an RDF is created with the anticipation of User Exit processing, the User Exit option on the Creation Options screen is set to YES. REPORT then requests the User Exit subroutine object code file parameters following field selection. If an existing RDF is modified to support User Exit processing, the User Exit option is specified during Report Modification. To specify the User Exit option, press PF8 from the Report ID Specification screen and supply the file parameters of the User Exit program on the resulting screen. User Exit processing can be removed from an RDF by deleting the User Exit filename during Report Modification.

NOTE

A Report Definition file constructed for User Exit processing cannot be used to print a report without the User Exit. If the user wishes to print the report with and without User Exit manipulations, a separate, additional RDF, which does not contain the User Exit file parameters, should also be constructed.

5.5.2 Constructing the User Exit Subroutine

REPORT supports User Exit subroutines written in the VS Assembler, BASIC, COBOL, and FORTRAN languages. The program must conform to the specific language requirements for an external subroutine, as described in the VS Assembler Language Reference, VS BASIC Language Reference, VS COBOL Reference or the VS FORTRAN Language Reference.

The User Exit subroutine does not directly manage the report print file, but controls the printing of each line through the arguments passed to REPORT. The subroutine can alter both the report format and content. Some arguments are passed to the subroutine only for informational purposes; other arguments control the print line when returned to REPORT. These arguments may have any name in the subroutine providing that they are returned in the order required by REPORT. The arguments passed to and returned from the User Exit subroutine are summarized as follows, in the order in which they must appear in an argument list.

Argument	Length	Description
Line Information	4 bytes	Each byte in the argument provides a particular description of the line about to be printed. Versions 3.2.2 and later of REPORT access all four bytes; earlier versions access only byte one. User Exit subroutines written for earlier versions of REPORT are compatible with later versions. Each byte and its meaning are summarized as follows; each byte contains a character value.

Byte	Description
1	Indicates the type of the line about to be printed. The line type is supplied to the subroutine by REPORT. Any changes made to the line type within the subroutine are ignored by REPORT. Byte 1 has one of the following character values indicating a particular line type.

Value	Line Type
P	Page heading or report title.
C	Column heading or subheading.
D	Detail line containing information from the data file record, formatted according to RDF specifications.
1-5	Control break line. The number indicates the control break level encountered as defined on the Control Fields Specification screen.
T	Control break totals line at the end of the report.
S	Report summary option line, including summary option heading.
E	Print line does not originate from REPORT. The "E" line type indicates that all print lines

Value	Line Type
	from REPORT have already been printed. The "E" line type can only arise if the user has set Byte 4 of this argument to "1".

Byte	Description
2	Indicates which line number, of a maximum of three, is being printed for the current record. In a report where only one line is printed per record, the value of this field is "1". In a report with more than one print line per record, the value can be "1", "2", or "3". When the line type is a report title or page heading ("P"), this field does not contain a meaningful value. The value indicating the multiple line number is passed to the subroutine by REPORT; any changes made to the multiple line number value by the User Exit subroutine are ignored by REPORT.
3	Indicates the end of data from the data file(s). The end of data field has a value of "0" until the last record in the input file has been read. A value of "1" indicates that the last data file record has been read. The end of data field value is passed to the subroutine by REPORT; REPORT ignores the value returned by the subroutine.
4	Indicates whether or not control should pass to the User Exit subroutine after the last report line has been printed. The default value of "0" returns control to REPORT; if the byte is set to "1" by the subroutine, control returns to the subroutine after each print line until the subroutine resets the byte to "0". If the byte is set to "1", the User Exit subroutine can print additional information at the end of the report. Unless this byte is set to "1", there is no way of inserting additional information or comments at the end of a report, as control automatically returns to REPORT when the last line has been printed. The Print Switch value (Argument 3 in the argument list) continues to influence whether or not page or column headings are printed when

Byte	Description
	a top-of-form is reached. A Print Switch value of "P" prints the headings; a value of "R" suppresses them. The User Exit subroutine defines a line to be printed and then sends it to REPORT for printing. Processing continues on a line-by-line basis until the subroutine resets the byte value to "0".

Argument	Length	Description
Print Line Count	2 bytes	Indicates the number of lines currently on the page, including the current line if it has been generated by the REPORT utility. This byte controls top-of-form processing. Thus, the User Exit subroutine must update the print line count whenever lines are added, deleted, or affected by a change in the control character of the print line. Failure to update the print line count results in irregular top-of-form processing. The argument data type is BINARY. Because BASIC has 4-byte integers, BASIC programs must ensure compatibility.
Print Switch	1 byte	Indicates the action REPORT should take with the print line. The value is set by the subroutine; each of the following values has the indicated meaning. REPORT sets a default value of "P". The Print Switch argument has the CHARACTER data type. Value — Line Type 0 — Omit printing the line. P — Print line and continue with report. R — Print line and return to the subroutine before processing the next print line. In this way, the user can insert comments or additional information in the middle of the report. The user must reset the Print Switch to P when printing the last of the inserted lines.
Print Line	134 bytes	Contains the 134 character current print line. The first two print line characters are the print control characters, described in the VS Principles of Operation. The User Exit subroutine can affect the line content by changing the value of this argument. In addition, blank lines can be inserted through the print control characters. If blank lines are inserted, the print line count must be updated.

Argument	Length	Description
Primary Record Area	Record length of primary data file	Contains the current record from the primary data file or view. Any fields on which reporting is not allowed contain blanks. This argument is passed to the subroutine by REPORT; changes made to the record by the subroutine do not affect either the record in the data file or the record viewed by REPORT. The field definition and record length in the subroutine should be consistent with those of the primary data file. In COBOL programs, the record description can be obtained from the Copylib function of CONTROL. (Refer to Chapter 2, CONTROL.) Because this argument corresponds to the primary data file rather than the report print file, the value of this argument does not necessarily change with every call to the User Exit subroutine. The value can be identical to that of a previous call if a multiple print line was defined in the RDF or if the current print line does not involve the data file, such as page or column heading lines or lines inserted into the report by the User Exit subroutine. This argument allows the user to use report field values in calculations.
Secondary Record Area	Record length of secondary data file	Contains the current record from the secondary data file or view if the report is based on two data files. This argument is passed to the subroutine by REPORT only if the RDF specifies a second data file. The argument description is identical to that of the primary record area in all other respects.

An example COBOL User Exit subroutine follows that illustrates the use of the subroutine arguments to alter the format and content of a report. The subroutine monitors the print file generated by REPORT that contains the department, name, address, hourly rate, and number of hours worked for an employee data file. The User Exit subroutine omits entries for employees not in department 50, prints a message on the print lines corresponding to those entries with an hourly rate exceeding $20, and prints a message at the end of the report.

```
000100 IDENTIFICATION DIVISION.
000200 PROGRAM-ID. FIXRPT.
000300*
000400*  THIS IS A SAMPLE USER EXIT PROGRAM FOR THE REPORT UTILITY.
000500*  THIS USEREXIT MODIFIES THE PRINTED REPORT IN THE FOLLOWING
000600*  MANNER.
000700*    IF A PERSON IN THE PRINTED REPORT IS NOT IN DEPARTMENT 50
000800*      THEN THE PRINTLINE IS OMITTED.
000900*    IF A PERSON IN DEPARTMENT 50 EARNS MORE THAN $20 PER HOUR,
001000*      THEN A MESSAGE IS PLACED ON THE PRINT LINE BEFORE IT
001100*      IS PRINTED.
001110*    AT THE END OF THE REPORT, THE USER EXIT PROGRAM ADDS A FEW
001120*      ADDITIONAL LINES OF TEXT.
001200*
001300 ENVIRONMENT DIVISION.
001400 CONFIGURATION SECTION.
001500 SOURCE-COMPUTER. VS2200.
001600 OBJECT-COMPUTER. VS2200.
001700*
001800 DATA DIVISION.
001900 WORKING-STORAGE SECTION.
002000 77  FIRST-TIME-INDICATOR PIC X VALUE "0".
002100     88  FIRST-TIME VALUE "0".
002200 77  NOT-FIRST-TIME PIC X VALUE "1".
002300 77  END-LINE-COUNT PIC 99.
002400*
002500 LINKAGE SECTION.
002600 01  LINE-INFO.
002700     02  LINE-TYPE   PIC X.
002800     02  LINE-NUMBER PIC X.
003000     02  END-OF-DATA PIC X.
003100         88  MORE-DATA VALUE "0".
003150     02  RETURN-AFTER-LAST PIC X.
003200 01  PRINT-LINE-COUNT USAGE IS BINARY.
003300 01  PRINT-SW PIC X.
003400 01  PRINT-LINE.
003500     02  PRINT-LINE-CC USAGE IS BINARY.
003600     02  PRINT-LINE-TEXT.
003700         03  FILLER PIC X(97).
003800         03  PRINT-LINE-MESSAGE-AREA PIC X(25).
003900         03  FILLER PIC X(10).
004000 01  RECORD-AREA.
004100     02  DEPARTMENT   PIC 99.
004200     02  NAME         PIC X(20).
004300     02  ADDRESS      PIC X(25).
004400     02  HOURLY-RATE  PIC 99V99.
004500     02  HOURS-WORKED PIC 99.
004600*
004700 PROCEDURE DIVISION USING LINE-INFO, PRINT-LINE-COUNT, PRINT-SW,
004800     PRINT-LINE, RECORD-AREA.
004900*
```

```
005000 PROCESS-A-PRINT-LINE.
005100     IF FIRST-TIME THEN PERFORM FIRST-TIME-PROCESSING.
005200     IF MORE-DATA AND LINE-TYPE = "D" THEN
005300         IF DEPARTMENT NOT = 50
005400             THEN PERFORM OMIT-THE-LINE,
005500         ELSE IF HOURLY-RATE > 20.00
005600             THEN PERFORM CHANGE-THE-PRINT-LINE.
005700     IF LINE-TYPE = "E" THEN PERFORM END-THE-REPORT.
005800     EXIT PROGRAM.
005900*
006000 FIRST-TIME-PROCESSING.
006100     MOVE NOT-FIRST-TIME TO FIRST-TIME-INDICATOR.
006200     MOVE "1" TO RETURN-AFTER-LAST.
006300     MOVE 1 TO END-LINE-COUNT.
006400*
006500 OMIT-THE-LINE.
006600     MOVE "O" TO PRINT-SW.
006700     SUBTRACT PRINT-LINE-CC FROM PRINT-LINE-COUNT.
006800*
006900 CHANGE-THE-PRINT-LINE.
007000     MOVE "THIS EMPLOYEE IS OVERPAID" TO PRINT-LINE-MESSAGE-AREA.
007100*
007200 END-THE-REPORT.
007300     IF END-LINE-COUNT = 1 THEN
007400         MOVE "THIS IS THE END OF THE REPORT" TO PRINT-LINE-TEXT,
007500         MOVE "R" TO PRINT-SW,
007510         ADD 1 TO END-LINE-COUNT,
007600     ELSE
007700         MOVE "**********************************" TO PRINT-LINE-TEXT,
007800         MOVE "P" TO PRINT-SW.
007900         MOVE "0" TO RETURN-AFTER-LAST.
```

5.5.3 REPORT Processing with a User Exit Subroutine

Once the RDF has been supplied with the file parameters of an existing compiled or assembled User Exit subroutine, the report can be printed through PF4 of the REPORT Utility menu. The report is printed according to the report format and content as modified by the subroutine. It is not necessary to link the User Exit subroutine to REPORT through the LINKER utility; REPORT automatically links to the subroutine without user interaction.

When the report is printed, the REPORT utility constructs each print line as defined in the RDF and passes the print line (along with the other arguments described in Section 5.5.2, Constructing the User Exit Subroutine), to the User Exit subroutine for further processing. After the subroutine returns the optionally modified print line, REPORT prints the line and constructs the next print line. All printing is carried out through the REPORT utility. When the last line has been placed in the print file, processing terminates and the print file is available. The immediate disposition of the print file is dependent on the user's print mode defaults. Refer to the VS Programmer's Introduction for a discussion of print mode defaults and the print queue.

5.6 A SAMPLE REPORT PROCEDURE

Although report definition and modification are necessarily interactive processes, REPORT function selection, report printing, and much of the file specification in the definition and modification phases can be automated through the VS Procedure language. A complete list of REPORT GETPARMs is given in Appendix A, File Management Utility GETPARMs; consult the VS Procedure Language Reference for details concerning procedure syntax.

For report definition and modification, REPORT does not request file parameters through GETPARMs unless default values are incorrect or nonexistent. However, since only the file names of the primary and secondary data files need be specified, the library and volume locations can be specified at print time through a procedure. The volume locations and any altering of default values for the data files' corresponding control files must be specified following file name specification. The specification of the volume name and alteration of any default value for the RDF can be supplied through a procedure immediately following field definition for report creation and immediately following report ID specification for report printing and modification.

The following procedure initiates report creation and prints the report. The procedure supplies the volume locations for the primary and secondary control files and overrides the default library for the secondary control file; supplies the volume name of the RDF whenever required (accepting the REPORT default library); and specifies the library and volume names of the primary and secondary data files. The user need only specify the file names of the primary and secondary data files, define the fields for the report, and supply the appropriate report definition options.

```
PROCEDURE
RUN REPORT
ENTER FUNCTION 2
ENTER CONTROL VOLUME=SYSTEM
ENTER CONTROL2 LIBRARY=MLHCTL1, VOLUME=SYSTEM
ENTER RPTDEF VOLUME=SYSTEM
ENTER FUNCTION 4
ENTER OPTIONS ID=LEELA, PAGE=40
ENTER RPTDEF VOLUME=SYSTEM
ENTER INPUT1 LIBRARY=MLHLIB, VOLUME=SYSTEM
ENTER INPUT2 LIBRARY=MLHLIB, VOLUME=SYSTEM
ENTER FUNCTION 16
RETURN
```

CHAPTER 6
INQUIRY

6.1 INTRODUCTION

While REPORT provides a formal means of data retrieval, INQUIRY facilitates interactive data file interrogation. INQUIRY enables the user to retrieve selected portions of data files. Thus, specific data fields can be displayed at the workstation or a subset of the input data file's records can be written to an output file. The fields or records retrieved by INQUIRY meet criteria specified by the user in the form of a query. The INQUIRY query is formulated in conversational English, eliminating the need for the user to learn a special query language. A data file interrogated by INQUIRY must have an associated control file.

INQUIRY processing involves the following steps, documented in the indicated sections.

- The data file to be interrogated must be selected, as must the type of interrogation desired. Refer to Section 6.2 for details.

- The particular information desired from the data file is defined in a query. Refer to Section 6.3 for a description of the necessary query components.

- When the user has successfully interrogated the data file, file parameters must be specified if INQUIRY creates an output data file. In addition, INQUIRY can write a procedure to automatically re-create the query results. If the user wishes to print a report on the data, INQUIRY can construct a Report Definition file from which the REPORT utility can print the report. The output options are described in Section 6.4.

- User workstation interaction can be significantly reduced through the VS Procedure language. Refer to Section 6.5 for details.

An overview of INQUIRY processing is provided in Figure 6-1.

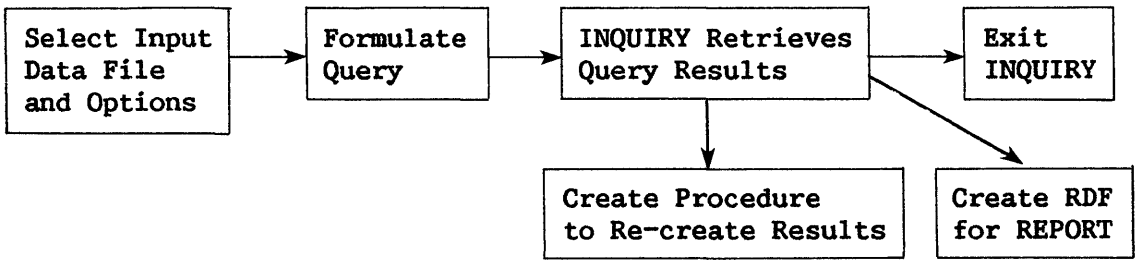


Figure 6-1. INQUIRY Processing

6.2 SELECTING THE INPUT OPTIONS

When the INQUIRY utility is run, the first screen requests the file parameters of the data file to be interrogated, as well as the ultimate disposition of the query results. The user must specify the file, library, and volume names of the data file. INQUIRY assumes that the data file has a corresponding control file with the same file name residing in the user's default control library on the default input volume. If no default input volume has been specified, INQUIRY assumes the control file resides on the same volume as the data file. If this is not the case, the user should set the Control File Change parameter to YES, overriding the default value of NO, to specify the actual control file parameters on a subsequent screen.

The user must also select the mode in which INQUIRY should open the data file. INQUIRY supports two open modes for data files: INPUT and SHARED. Indexed data files may be opened in either mode; consecutive data files must be opened in Input mode. If Input mode is specified, INQUIRY can only open the data file if no other program is currently updating the file. However, two or more users can open the same file in Input mode through INQUIRY, since INQUIRY only reads and does not update the data file. Thus, Input mode ensures that the contents of the data file do not change during INQUIRY processing. If Shared mode is specified, INQUIRY can open the indexed file even if it has been previously opened in Shared mode by another program such as DATENTRY. However, because the file contents can be updated by other programs during INQUIRY processing, two identical queries processed at separate times may have different results.

In addition, the user must specify the output option for the query. INQUIRY supports two output options: DISPLAY and EXTRACT. The Display option displays on the workstation the values of specified fields contained in a record meeting the query's criteria. The Extract option writes the entire record, as defined in the control file, to a file if the record meets the query's criteria. Records not meeting the specified criteria are omitted from the file. The output file is named after an Extract query is processed. A separate file is created for each query. The default response is DISPLAY; the user can override the default value on the Input Specification screen if an output file is desired.

6.3 FORMULATING THE QUERY

Once the input file or view has been selected and the control file specified (if necessary), the user formulates the query on the Query Specification screen. A query contains two parts: the list clause and the relation clause. The list clause, which specifies the fields whose values are sought in the Display option, is described in Section 3.1. The relation clause, which indicates the criteria on which the field values (for the Display option) or records (for the Extract option) are selected, is described in Section 6.3.2. The syntactical requirements of a complete query and example queries are given in Section 6.3.3.

6.3.1 The List Clause

Because the entire record satisfying the relation clause is written to the extracted file, the list clause has no meaning for the Extract option. Therefore, the list clause can be omitted from a query written for the Extract option. If a list clause is included in a query for the Extract option, it is ignored by the INQUIRY utility.

The list clause has the following structure.

```
VERB   FIELDNAME-1  [FIELDNAME-2  ...  FIELDNAME-N]
```

The following verbs are recognized by INQUIRY: LIST, DISPLAY, FIND, WRITE, PRINT, SHOW, and GIVE. However, INQUIRY does not associate a distinct meaning with each verb; the user can select the verb that best fits the particular query. The field names are either control file field names or field aliases. (Refer to Chapter 2, CONTROL, for details). Those field names or aliases contained in the list clause are displayed as a result of the query with the Display option.

6.3.2 The Relation Clause

The relation clause has the following structure.

```
FIELDNAME-X   RELATIONAL   { NUMERIC VALUE   }  [ CONNECTOR   RELATION ]
              OPERATOR     { FIELDNAME-Y     }  [           CLAUSE   ]
                           { "LITERAL"       }
                           { HEXADECIMAL VALUE }
```

The field name (or alias) of the relation clause is identified solely by its location prior to a relational operator. Thus, field names positioned before the relation clause field name are considered part of the list clause, if present. If the query contains no list clause, multiple field names preceding a relational operator result in an error. Thus, it is not possible to specify a criterion for two fields in the same relation clause. Any field forming part of the relation clause that is not mentioned in the list clause is also displayed when the query is processed by the Display option. The Display option displays relation clause fields following list clause fields.

Multiple criteria are specified through multiple relation clauses attached to the query through the connector. Relational operators include EQ, NE, LT, GT, LE, GE, the associated mathematical representations (=, >, etc.), or the English equivalents (EQUAL TO, GREATER THAN OR EQUAL TO, etc.). The field contents can be compared to either a numeric value, another field value, a literal value, or a hexadecimal value. A numeric field can be compared to either another numeric field or a numeric literal. A character field can be compared to another character field, a hexadecimal field, or a literal. A numeric value is a digit or sequence of digits that can contain a sign and/or a decimal point. Literal values include any displayable, constant value, and must be enclosed in quotes in the query.

The name or alias of any field in the record can also be compared to the contents of another field by specifying a field name or alias after the relational operator. In this way, the user could, for example, retrieve only those records for which the total number of hours worked exceeded the number of regular (nonovertime) hours worked. A hexadecimal value used for comparison in the relation clause is specified in the form X'hh...h', where 'h' represents one of an even number of hexadecimal values in the range 0-9 and A-F. Multiple criteria are specified by appending a connector and another relation clause to the query. The connector must be either AND or OR. If AND is used as the connector, both criteria must be true for the specified field(s) or record to be selected; an OR connector displays the field(s) specified in the list clause or writes the record to a file when either criterion is true. If the relation clause contains multiple connectors, all AND relations are evaluated first. The AND result is then treated as a single criterion to be used by the OR connectors.

6.3.3 Complete Queries

The following example illustrates the query results for typical queries for both the Display and Extract options.

Query	DISPLAY Result	EXTRACT Result
LIST A B C D GT 'Leela'	The values of the A, B, C, and D fields are displayed for those records with D field values after 'Leela'.	Those records containing values after 'Leela' in the D field are written to a file.
DISPLAY A A GE X'00'	Those values of the A field greater than or equal to hexadecimal '00' (i.e., all values) are displayed.	Those records containing A field values greater than or equal to hexa-decimal '00' (i.e. , all values) are written to a file.
RATE LESS THAN 4.00	Error. List clause required.	The records for employees earning less than $4.00/hour are written to a file.

Query	DISPLAY Result	EXTRACT Result
LIST A GT 1 AND B GT 1 OR B EQ 0	The values of the A and B field for which A and B both exceed 1 and for which A has any value and B equals 0 are displayed. Note that the case of A=0 and B=0 is allowed. Also note that while no fields are mentioned in the list clause, the A and B fields are displayed due to their specification in the relation clause.	Those records in which the A and B fields both exceed 1 and those records with any value of A and B values of 0 are written to a file. Note that the case of A=0 and B=0 is allowed.
LIST A GT 1 AND B GT 1 OR A GT 1 AND B EQ 0	The values of the A and B fields for which A and B are both greater than 1 and for which A is greater than 1 and B equals 0 are displayed. Note that the case of A=0 and B=0 is not allowed. Also note that while no fields are mentioned in the list clause, the A and B fields are displayed due to their specification in the relation clause.	Those records for which the A and B fields both exceed 1 and those records with A values greater than 1 and B values of 0 are written to a file. Note that the case of A=0 and B=0 is not allowed.

The preceding description of a query describes only the necessary components of a query. Since INQUIRY ignores words it does not recognize or consider necessary, the user can formulate the query in a readable, English sentence, with punctuation inserted as desired. The following queries produce the same results as those in the previous examples.

```
PLEASE LIST THE A, B, AND C FIELDS FOR WHICH THE D FIELD IS GREATER
    THAN 'Leela'.  THANK YOU.

DISPLAY THE A FIELD FOR VALUES OF A GE X'00'.

EXTRACT THOSE RECORDS FOR EMPLOYEES WHO HAVE AN HOURLY RATE LESS THAN
    4.00 PER HOUR.

LIST THOSE VALUES FOR WHICH A IS GREATER THAN 1 and B IS GREATER THAN
    1 OR FOR WHICH B IS EQUAL TO 0.

AT YOUR EARLIEST CONVENIENCE, DISPLAY FIELDS WITH A GREATER THAN 1 AND
    B GREATER THAN 1 OR FIELDS WITH A GREATER THAN 1 AND B GREATER
    THAN 0.
```

The query is entered on a screen containing seven rows of pseudoblanks. Instructions for query construction can be obtained during INQUIRY processing by pressing PF14 from the Query Specification screen. A list of field names and aliases associated with the data file can be obtained during INQUIRY processing by pressing PF15. The Query Specification screen can then be redisplayed unchanged by pressing PF1. Pressing PF2 enables the user to specify a character literal in lowercase; pressing PF2 from lowercase mode returns the user to uppercase mode. For the Display option only, the user can optionally specify a title to appear on the workstation screen with the query results. When the user presses ENTER after specifying the query, INQUIRY displays the query as interpreted by the utility. If the query is incorrect, the user can modify the query and again press ENTER to view the INQUIRY interpretation. When the user is satisfied with the query, pressing PF16 initiates INQUIRY processing of the query. The user can also return to the Input Specification screen by pressing PF1 from the Query Specification screen.

6.4 MANAGING INQUIRY OUTPUT

For the Display option, INQUIRY displays the query results on the workstation screen through a link to the DISPLAY utility. The user can then view and/or print the results through DISPLAY. Once the results have been viewed, control is returned to INQUIRY by pressing PF16. A message is displayed if no records satisfy the query criteria; pressing ENTER to acknowledge the message resumes INQUIRY processing.

The user must specify the file, library, and volume names of the output file if the Extract option was selected. INQUIRY supplies a default number of records for file creation, based on the number of records in the interrogated data file. This default value can be overridden by the user, if appropriate. Ordinarily, INQUIRY constructs the extracted data file with the same file organization as the interrogated data file. However, for indexed data files only, the user can opt to create a consecutive file instead, by overriding the NO default value of the CONSEC parameter. A consecutive file requires less space to create. However, the Consecutive File option should only be selected if further processing of the file does not require indexed organization. The file, library, and volume names and number of records created in the output file are displayed on a subsequent screen.

Once the query results have been displayed at the workstation or written to a file, INQUIRY displays an End of Query screen that presents the user with four options, described as follows.

- The user can construct another query that interrogates the same data file or a different data file. Refer to Section 6.4.1 for details.

- The user can create a procedure that automatically recreates the previous query results. Refer to Section 6.4.2 for details.

- The user can create a Report Definition File based on the query results. Refer to Section 6.4.3 for details.

- The user can exit the INQUIRY utility by pressing PF16.

6.4.1 Formulating Additional Queries

If PF1 is selected from the End of Query screen, INQUIRY redisplays the Input File Specification screen, with the entries from the previous specification supplied as default values. The user can then alter the responses as required and proceed to the Query Specification screen, where the previous query is displayed as a default value. The user can then accept or redefine the query and continue INQUIRY processing. Thus, the user can repeatedly interrogate many files in a single INQUIRY session, with specification of related queries facilitated by the default responses. If the Extract option was selected, the user specifies an output file for each query.

6.4.2 Saving the Completed Query

If PF2 is selected from the End of Query screen, INQUIRY constructs a procedure through which the query results can be re-created without user interaction. Thus, the procedure runs the INQUIRY utility, specifies the query, and displays or extracts the query results. The query can be displayed for verification and/or modification before interrogating the data file, if desired. All query options selected on the Input Specification, Control File Specification, and Query Specification screens in the current INQUIRY session are specified in the output procedure.

When this option is selected from the End of Query screen, the user must specify the file, library, and volume names of the procedure to be created by INQUIRY on the Procedure Specification screen. The default procedure action displays the query for verification and/or modification before interrogating the data file. If the value of the DISPLAY parameter is changed to NO, the query is automatically processed when the procedure is run. The query results are then either immediately displayed or placed in a user-specified file location. The procedure can either terminate processing or return the user to the End of Query screen when the query has been processed. The default value of the END RUN parameter, NO, displays the End of Query screen; if the value of END RUN is changed to YES, INQUIRY execution is terminated when the query has been processed.

Pressing ENTER from the Procedure Specification screen creates the procedure and returns the user to the End of Query screen. Alternatively, the user can return to the End of Query screen without creating a procedure by pressing PF1.

To run the output procedure, the user need only specify the file, library, and volume names of the procedure in the RUN option of the Command Processor, or in a RUN statement of another procedure. If the Display option is operative, the query results are displayed and the END RUN parameter value determines the end of query processing action. If the Extract option was selected, the previous output file location is assumed. If the file already exists, the standard file respecification GETPARM appears, allowing the user either to scratch the existing file by pressing PF3 or supply new file parameters.

6.4.3 Creating a Report Definition File

A Report Definition File (RDF) for use with the REPORT utility can be created based on the query results. For a description of an RDF, refer to Chapter 5, REPORT. INQUIRY provides the user with an automatic method of report creation as the RDF created by INQUIRY can be printed by REPORT without modification. If the Display option was used in the query, only those fields requested in the query are placed in the RDF. All data file fields are placed in the report if the Extract option is used. Because the INQUIRY selection criteria are not compatible with REPORT data limits, the RDF reflects the entire input data file. One method of creating an RDF that reflects the INQUIRY selection process would be to run the INQUIRY utility twice. In the first INQUIRY session, the user should select the Extract option to create an output data file containing only the desired records. A second INQUIRY session using the previously extracted data file as the input file could then create the RDF.

When the Create Report Definition File option is selected by pressing PF3 from the End of Query screen, INQUIRY requests the file, library, and volume names of the RDF to be created. INQUIRY supplies the default library name of an RDF (the User ID concatenated with "RPT"). This value may be overridden, if appropriate. Pressing ENTER following file location specification creates the RDF; INQUIRY returns to the End of Query screen without creating an RDF if PF1 is pressed.

The file name specified for the RDF is considered both the Report ID and the Report Title. The RDF created by INQUIRY contains REPORT default values for all Report Definition options and can be modified through REPORT, as described in Chapter 5, REPORT. To ensure reasonable results, INQUIRY limits the number of fields included in the report to 80. Any additional fields included in the query results are ignored for report definition. In addition, once the print line has reached capacity, INQUIRY assigns subsequent fields a sequence number of 99. As described in Chapter 5, REPORT, a 99 sequence number includes the field in the RDF, but omits the field from the print line. The user can alter INQUIRY's default actions through the REPORT Modify option before printing the report.

6.5 A SAMPLE INQUIRY PROCEDURE

Although INQUIRY can create a procedure that reproduces the query results, the user can automate only those portions of the INQUIRY processing appropriate to the individual application through the VS Procedure language. Workstation interaction for the Extract option can be completely suppressed. All phases of INQUIRY processing under the Display option can be automated except for the workstation display of the query results. A complete list of the INQUIRY GETPARMs is given in Appendix A, File Management Utility GETPARMs; consult the VS Procedure Language Reference for details concerning procedure syntax.

The Query Specification screen displayed during normal INQUIRY processing is not a GETPARM screen. Thus, the query cannot be specified in the procedure in a method parallel to the method described in Section 6.3. INQUIRY contains a hidden GETPARM, identified by the QUERY prname, through which query specification can be accomplished and the usual Query Specification screen suppressed.

The first keyword in the QUERY GETPARM, DISPLAY, determines whether the Query Specification screen should appear for verification of the query supplied through the procedure. The default value, YES, displays the Query Specification screen with the procedure-supplied query displayed. If the value of the DISPLAY keyword is changed to NO, INQUIRY attempts to process the query as specified in the procedure. The Query Specification screen then only appears if the procedure-supplied query has an error. The title for displayed query results can be supplied through the TITLE keyword.

The remainder of the QUERY GETPARM specifies the query. For procedure purposes, the seven-row pseudoblank area on the Query Specification screen is replaced by 14 keywords, where each line is described by two keywords. The query is specified by supplying values enclosed in quotes to as many keywords as are necessary to form the query. The total query must conform to the query standards detailed in Section 6.3.

A sample procedure follows. The procedure runs the INQUIRY utility; specifies the input data file in SHARED mode with a control file change and the INQUIRY Display option; supplies the file parameters of the alternate control file; supplies the query for immediate display processing with a title; creates and names a Report Definition file; runs the REPORT utility; selects the REPORT Print function; and prints the report.

```
PROCEDURE
RUN INQUIRY
ENTER INPUT FILE=LEELA, LIBRARY=MLHLIB, VOLUME=SYSTEM, CHANGE=YES,
      MODE=SHARED, OPTION=DISPLAY
ENTER CONTROL FILE=LEELA, LIBRARY=MLHCTL1, VOLUME=SYSTEM
ENTER QUERY DISPLAY=NO, TITLE=TEST, LINE1A="LIST A",
      LINE2B="FOR A LT 'Z'"
ENTER EOJ 3
ENTER RPTDEF FILE=A, VOLUME=SYSTEM
ENTER EOJ 16
RUN REPORT
ENTER FUNCTION 4
ENTER OPTIONS ID=A, DEVICE=PRINTER
RETURN
```

CHAPTER 7
CONDENSE

7.1 INTRODUCTION

The CONDENSE utility enables the user to construct a file with a single record type from a file containing multiple record types. Since the REPORT utility accepts only files with a single record type, CONDENSE facilitates reporting on files with multiple record types.

CONDENSE requires a control file for each record type in the input data file. The user can select which fields in each record type are transferred to the condensed data file. In addition, the user can entirely omit undesired record types, or only include records of a given type that meet specified criteria. The output data file has the structure of the sum of each contributing control file, modified by omitted fields. In addition to the output data file, CONDENSE creates a control file corresponding to the output file structure. The output data file can be accessed by any of the File Management Utilities when used with the output control file; the output data file can also be used in any user application.

CONDENSE does not directly build the output data file, but processes the multiple record type input file through a buffer. The buffer has the structure of the output data file. As each record in the input file is processed, it is placed in the area of the buffer corresponding to its record type. The buffer contents are transferred to the new condensed file only when specified by the user, allowing user design of the output file.

CONDENSE processing involves the following steps, documented in the indicated sections.

- The record types are identified by control files and additional identifying criteria. The user also selects the fields to be transferred and specifies any record selection criteria. From this information, CONDENSE constructs a parameter file that governs output file creation. Consult Section 7.2 for details.

- Previously created parameter files can be altered. Consult Section 7.3 for more information.

- Once an appropriate parameter file has been built, CONDENSE creates the condensed data file based on the criteria specified in the parameter file. A corresponding control file is also constructed. Refer to Section 7.4 for details.

- Once the condensed file and corresponding control file have been created, a report can be made on the file through the REPORT utility. Refer to Section 7.5 for details.

- Through a User Exit subroutine, the user can define selection criteria not included in the CONDENSE utility and/or modify the condensed data file's contents. Refer to Section 7.6 for a discussion of the processing requirements.

- Workstation interaction can be minimized through the VS Procedure language. Refer to Section 7.7 for details.

An overview of CONDENSE processing is provided in Figure 7-1.

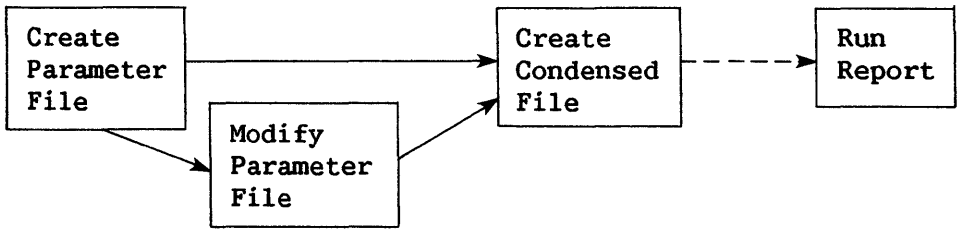


Figure 7-1. CONDENSE Processing

When CONDENSE processing begins, the file, library, and volume names of the parameter file must be specified on the first screen, referred to as the Parameter File Specification screen. The parameter file's library defaults to the value obtained by concatenating the User ID with "COND". Pressing ENTER accesses the CONDENSE Function menu; pressing PF16 terminates CONDENSE processing. The following options are available from the CONDENSE Function menu.

PF Key	Function
2	Create a Parameter File
3	Modify a Parameter File
4	Create a Condensed File
5	Run REPORT
16	Return to the Parameter File Specification Screen

7.2 CREATING A PARAMETER FILE

The parameter file contains all the information required to transform the multiple record type input file into the desired single record type output condensed file. The user specifies a control file for each record type that is to be included in the output file. Record types not identified by a control file are omitted from the data file. Because control files are indexed, CONDENSE defines a single output record that corresponds to the alphabetical concatenation of each specified control file represented in the multiple record type file. Note that, for this reason, the control files and field names must be unique. Within each record type area, the fields are placed in alphabetical order. Note that while the output data and control files have this structure, a report on the condensed file generated with the REPORT default field sequence places all fields in the file in alphabetical order with respect to the entire condensed data file.

Parameter file creation involves the following steps, documented in the indicated subsections.

- A record is uniquely identified and options are selected that control the transfer of the buffer contents to the output file when the record type is encountered. Defining record types is discussed in Section 7.2.1.

- Once a record type has been defined, the user selects which fields within the record type are to be included in the condensed file. Thus, CONDENSE alternately displays Record Type Selection and Field Selection screens until all record types have been defined. Field selection is treated in Section 7.2.2.

- Once all record types and fields have been defined, the parameter file is completed by naming the input and output files, as discussed in Section 7.2.3.

7.2.1 Record Type Definition

When PF2 is pressed from the CONDENSE Function menu, CONDENSE displays a Record Type Selection screen on which the user identifies a desired record type and specifies if and how the buffer contents should be transferred to the condensed data file when the record type is encountered. Since CONDENSE places the record types in the condensed data file in alphabetical order of control file names, the order in which the user specifies each record type is irrelevant.

On the Record Type Selection screen, the user must first provide the file parameters of the control file corresponding to the record type being specified. The user must also uniquely identify the record type by providing identification criteria for one or more bytes in the record. Up to ten criteria can be specified on the Record Type Selection screen. These criteria can also be used to select ranges of values within a record type. Thus, the user can include in the condensed file only those records of a given type that meet specific criteria merely by specifying stricter criteria than required to identify the record type.

The user indicates the criteria through four parameters: BYTE LOCATION, RELATIONAL OPERATOR, CHARACTER, and CONNECTOR. One criterion is composed of a BYTE LOCATION compared to a CHARACTER value through a RELATIONAL OPERATOR. If more than one criterion is required to uniquely identify the record type or to eliminate undesired record type ranges, additional criteria are linked through the CONNECTOR. Each parameter is described as follows.

BYTE LOCATION	Refers to the input record position tested by CONDENSE. The BYTE LOCATION must be specified as a number between 1 and the record length of the record type.
RELATIONAL OPERATOR	Specifies the arithmetic relationship for the test. Valid RELATIONAL OPERATORS are EQ (equal), NE (not equal), GT (greater than), LT (less than), GE (greater than or equal), and LE (less than or equal).
CHARACTER	Refers to the ASCII or hexadecimal value to which the value in the BYTE LOCATION is compared through the RELATIONAL OPERATOR. If one character is entered, the value is interpreted as an ASCII character; if two characters are entered, the value is treated as a hexadecimal value. The ASCII value must be entered into the left position of the two screen locations provided for CHARACTER entry.
CONNECTOR	Links multiple criteria. An AND value yields a positive test when the linked criteria are both true; an OR value yields a positive result if either criterion is true. Up to ten criteria can be linked for a given record type.

A typical set of criteria to identify a record type might be: 10 GE 4 AND 51 EQ 0A.

The remaining two parameters on the Record Type Selection screen, WRITE and RESET, control the creation of the condensed file. As each record type is encountered in the input data file, it is transferred to the area in the buffer reserved for the given record type. CONDENSE processes consecutive input files in sequential order and indexed input files in primary key sequence. The values of WRITE and RESET for the record type being processed determine whether the current buffer contents are written to the output file and whether the previous record type values remain in their buffer areas for the transfer. Thus, while the entire buffer is transferred to the output data file, the decision to transfer the buffer contents is made on a record-type-by-record-type basis. The meaning and allowed values of each parameter are summarized as follows.

WRITE Controls the transfer of the buffer contents to the output file when this record type is processed. The three possible values of WRITE, described below, determine the relationship of writing the buffer contents to the output file to the transfer of the selected fields of the current record to the buffer area. The default value of WRITE is NO.

NO The buffer contents are not written to the output file at this time. However, the selected fields of the current record are written to the buffer area, overwriting any previous entry.

BEFORE The buffer contents are written to the output file before the selected fields of the current record are moved to the buffer area.

AFTER The buffer contents are written to the output file after the selected fields of the current record are moved to the buffer area.

RESET Controls the clearing and reinitialization of the entire buffer. Fields from each specified record type remain in the output buffer until overlaid by other fields of the same record type, or until the entire buffer is cleared through the RESET parameter. The three possible values of RESET determine the timing of the buffer initialization with respect to the transfer of the selected fields of the current record of the given type to the buffer. The default value of RESET is NO. Each RESET value is explained as follows.

NO The buffer is not to be reinitialized when this record type is encountered.

BEFORE The buffer is reinitialized before the selected fields of the current record of this type are moved to the buffer area.

AFTER The buffer is reinitialized after the selected fields of the current record of this type are moved to the buffer area.

The user controls the structure of the output file by selecting the appropriate combination of WRITE and RESET values. Figure 7-2 illustrates the relative processing of the WRITE and RESET values with respect to the tranfer of the record to the buffer.

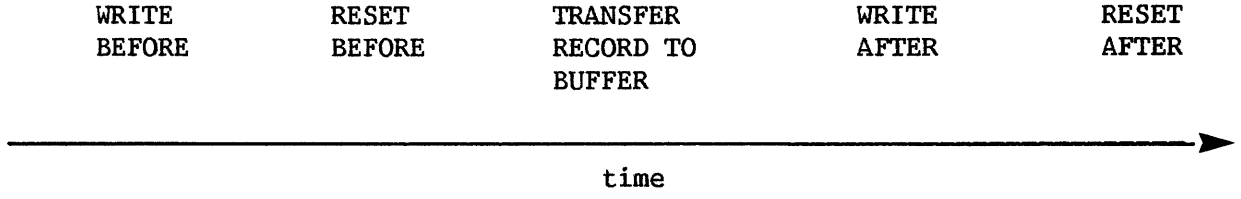
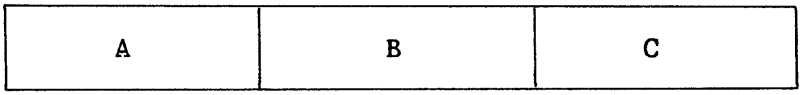


Figure 7-2. Relative Processing of WRITE and RESET

Thus, if a WRITE and RESET are both to be processed either before or after the transfer of the record to the buffer, the WRITE always occurs before the RESET. Combinations of WRITE=BEFORE and RESET=AFTER, and WRITE=AFTER and RESET=BEFORE, are allowed and are processed exactly as illustrated in Figure 7-2. For example, WRITE=AFTER and RESET=BEFORE results in the transfer of the buffer containing only the current record of the given type (all other record type values having been erased by the RESET=BEFORE) to the output file. Thus, WRITE and RESET typically have identical values, with mixed values useful in unusual applications.

The use of WRITE and RESET to structure the output file is illustrated in the following three examples. Assume a consecutive data file contains three record types A, B, and C, represented by control files A, B, and C, respectively. Thus, the output buffer has the following structure.



If one also assumes that the data file contains the following order of records:

A1 B1 B2 C1 C2 C3 A2 B3 C4 A3 C5 A4,

then Figures 7-3, 7-4, and 7-5 illustrate the results of differing WRITE and RESET values. Each figure contains a representation of the buffer contents as each record in the input data file is processed, and indicates when the buffer is cleared and when the buffer contents are transferred to the output data file.

	RESET	WRITE
A	NO	NO
B	NO	NO
C	AFTER	AFTER

INPUT FILE PROCESSING (TIME)

BUFFER			OUTPUT FILE
A1			
A1 B1			
A1 B2			
A1 B2	C1	WRITE Record 1 →	A1 B2 C1
RESET			
	C2	WRITE Record 2 →	C2
RESET			
	C3	WRITE Record 3 →	C3
RESET			
A2			
A2 B3			
A2 B3	C4	WRITE Record 4 →	A2 B3 C4
RESET			
A3			
A3	C5	WRITE Record 5 →	A3 C5
RESET			
A4			

Figure 7-3. WRITE and RESET Example I

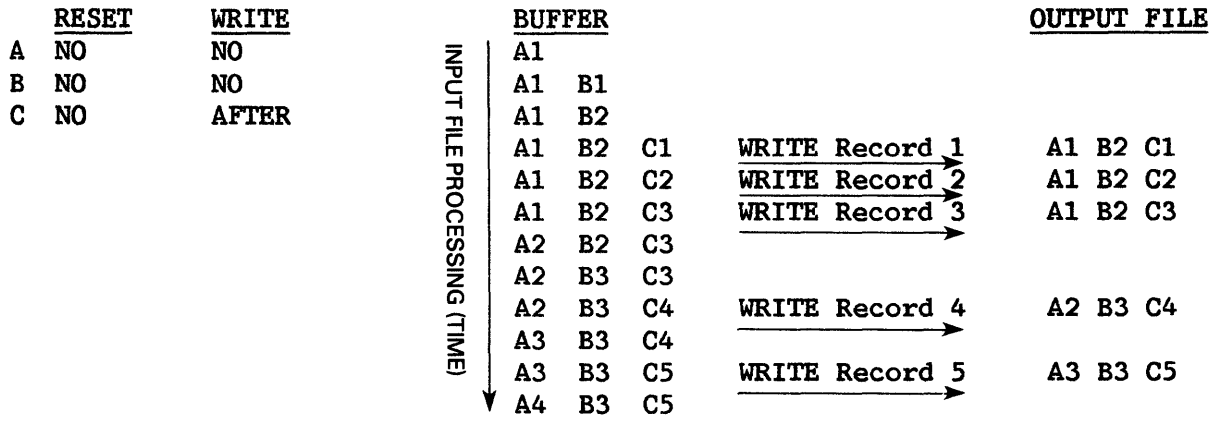


Figure 7-4. WRITE and RESET Example II

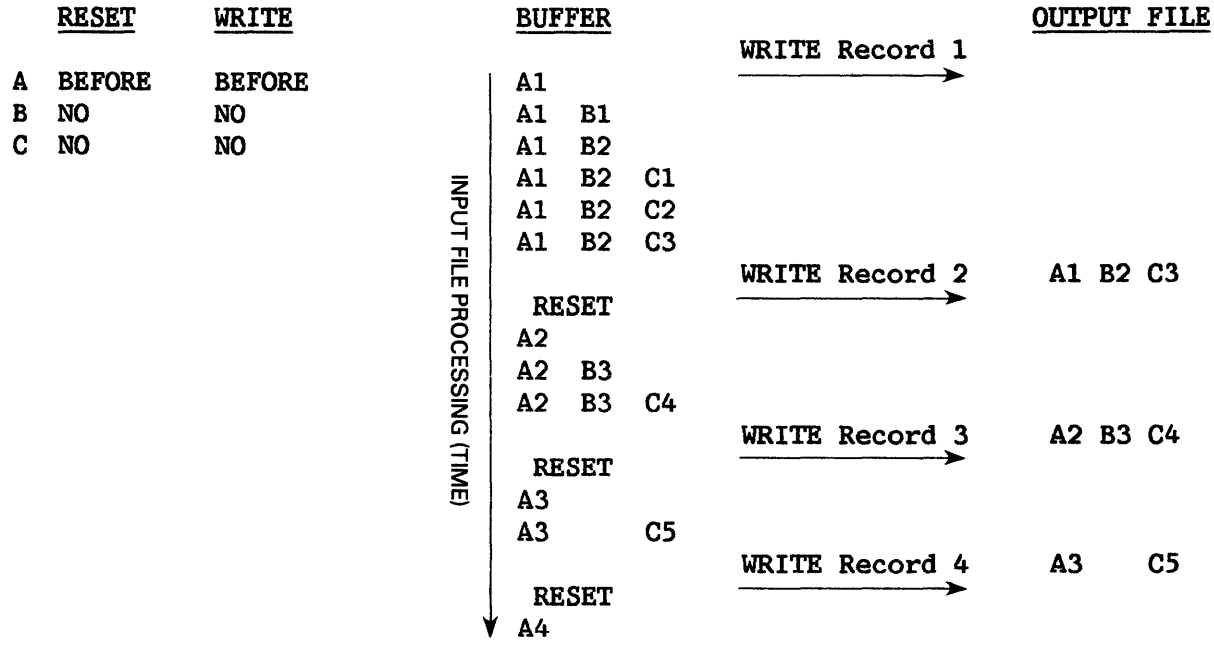


Figure 7-5. WRITE and RESET Example III

Information concerning the record type options can be accessed during CONDENSE processing by pressing PF14; pressing ENTER from the resulting information screen returns the user to the Record Type Selection screen. If all record types have already been defined, pressing PF16 begins the file definition process without processing any information currently entered on the screen. Pressing ENTER enables the user to select fields for the given record type on the Field Selection screen.

7.2.2 Field Selection

Once the record type options for a given record type have been entered on the Record Type Selection screen, the Field Selection screen is displayed, enabling the user to select which fields in the record type are to be included in the output file. The field names in the control file are displayed in alphabetical order. The field is included in the output file if a non-blank character is entered next to the field name. The selected fields are placed in the file in alphabetical order within the record type area. Thus, the file constructed from control file PERSONAL, with selected fields NAME, SALARY, and EMPID, and control file PUBLIC, with selected fields ADDRESS and TELNO, is stuctured as follows.

EMPID	NAME	SALARY	ADDRESS	TELNO

Two additional types of fields can be defined on the Field Selection screen: COUNTER and SUM fields.

COUNTER — Accumulates the total number of input records of this type selected since the last RESET. The COUNTER field is four bytes long (PIC S9(7) COMP). This field is named by the user and is placed alphabetically in the record type area of the output file. Only one counter field can be specified for a given record type.

SUM — Accumulates the values of a specified numeric field in this record type. The SUM field is eight bytes long (PIC S(15) COMP). A SUM field is specified by placing a plus sign (+) next to the field name to be summed. The original value of the designated sum field is replaced with the sum of that field's contents as each input record of the current type is processed. More than one sum field can be defined per record type. However, a numeric field specified by the user as a sum field retains its input control file field name and cannot be renamed by the user.

At least one field must be selected from those listed; if a counter field is selected, an additional data field must also be specified. Pressing ENTER completes specification for the given record type and displays the Record Type Selection screen on which another record type can be defined or from which the user can terminate record type definition. Pressing PF1 from the Field Selection screen displays the Record Type Selection screen, but does not include the given record type definition in the parameter file.

7.2.3 File Specification

Following record type and field selection, CONDENSE requests the file parameters of the input and output files, along with other information used in constructing the condensed file. Thus, once record type definition has been completed by pressing PF16 from a Record Type Selection screen, CONDENSE displays the Data File Specification screen. On this screen the user can provide the file parameters of the input multiple record type file and those of the output condensed file. An indexed condensed file is created if a key field name is supplied on the Data File Specification screen; if the key field parameter is left blank, a consecutive condensed file is created.

Note that the condensed file is named, not created, at this time. Thus, it is not necessary to supply the file parameters of the input and output files in the parameter file. If values are not supplied for the input file, output file, and control file parameters at this time, CONDENSE solicits the information through GETPARM requests when the condensed file is created. However, the GETPARM request does not enable the user to supply a key field name for the output condensed file, so a field name must be provided on the Data File Specification screen if an indexed condensed file is desired. A single parameter file can thus be built to transform several multiple record type files with the same file structure into consecutive condensed files.

Pressing ENTER from the Data File Specification screen accesses the Output Control File Specification screen; pressing PF16 presents the Parameter File Maintenance menu, described in Section 7.3.

The file parameters of the output control file corresponding to the condensed file can be supplied on the Output Control File Specification screen. The output control file's library defaults to the typical control file library name: the user ID concatenated with "CTL".

Two additional CONDENSE options can also be selected from the Output Control File Specification screen: User Exit subroutine processing and a PAD field to extend the condensed file's record length. A PAD field of blanks is placed at the end of each record in the condensed file if the desired field length is supplied. A PAD field of up to 999 bytes can be defined; a default value of 0 bytes is provided so that the field is only added to the condensed file when desired. A PAD field is useful if the addition of extra fields in the condensed file is anticipated. If the file parameters of a previously created User Exit subroutine object file are supplied, CONDENSE automatically links to the subroutine as described in Section 7.6.

When ENTER is pressed from the Output Control File Specification screen, the Parameter File Maintenance menu is displayed, allowing the user to modify any erroneous components of the parameter file.

7.3 MODIFYING A PARAMETER FILE

When PF3 is pressed from the CONDENSE Function menu, or when file definition has been completed during parameter file creation, the Parameter File Maintenance menu is displayed. Through this menu, the following operations can be performed, as documented in the indicated sections.

- A record type definition can be added to the parameter file. (Refer to Section 7.3.1.)
- The record type options and selected fields for a specific record type can be altered. (Refer to Section 7.3.2.)
- A record type definition can be removed from the parameter file. (Refer to Section 7.3.3.)
- The file parameters and options selected for CONDENSE input and output files can be changed. (Refer to Section 7.3.4.)
- Control is returned to the CONDENSE Function menu by pressing PF16.

7.3.1 Adding a Record Type

If PF2 is pressed from the Parameter File Maintenance menu, the Record Type Selection screen is displayed, allowing the user to specify a record type to be included in the condensed file. Once the record type options have been selected, pressing ENTER allows the user to define fields on the Field Selection screen. Pressing ENTER from the Field Selection screen includes the record type definition alphabetically in the parameter file and returns control to the Parameter File Maintenance menu. Pressing PF16 from the Record Type Selection screen cancels record type definition and displays the Parameter File Maintenance menu.

7.3.2 Modifying a Record Type

If PF3 is pressed from the Parameter File Maintenance menu, a Record Type Identification screen is displayed, indicating that modify mode has been selected. When the file parameters of the control file corresponding to a record type definition are entered, the Record Type Selection screen for the specified record type is displayed. Record Type options can then be modified. When the Record Type options are altered (if necessary), pressing ENTER displays the File Selection screen on which previous field selections can be modified. Pressing ENTER then displays the Record Type Identification screen from which another record type definition can be modified or control returned to the Parameter File Maintenance menu.

7.3.3 Deleting a Record Type

When PF4 is pressed from the Parameter File Maintenance menu, a Record Type Identification screen is displayed, indicating that delete mode has been selected. The record type definition is deleted if the file parameters of the control file representing the record type are entered; pressing PF16 returns control to the Parameter File Maintenance menu without deleting the record type definition.

7.3.4 Modifying File Definitions

If PF5 is pressed from the Parameter File Maintenance menu, the Data File Specification screen is displayed, allowing the user to modify the input multiple record type file parameters, output condensed file parameters, or the key field to the condensed file. Pressing ENTER displays the Output Control File Specification screen, on which the output control file and User Exit subroutine file parameters can be renamed and the PAD field redefined. Pressing PF16 from the Data File Specification screen returns the user to the Parameter File Maintenance menu; pressing ENTER from the Output Control file Specification screen displays the Parameter File Maintenance menu.

7.4 CREATING A CONDENSED FILE

When PF4 is pressed from the CONDENSE Function menu, the condensed data file and corresponding control file are constructed according to the parameter file specifications. If all file name, library, and volume locations have been correctly supplied in the parameter file, no user interaction is necessary, and the CONDENSE Function menu is displayed when the condensed data and conrol files have been created.

If the parameter file contains incorrect or no file definitions, the file parameters of the output control file, input multiple record type file, and/or output condensed file are sequentially solicited through separate GETPARM requests. Note that a User Exit subroutine cannot be specified at this time. From any such GETPARM request, the user can return to the CONDENSE Function menu by pressing PF1 without creating the condensed file. If all necessary GETPARM requests are fulfilled, the condensed data and control files are created, and control is returned to the CONDENSE Function menu.

7.5 REPORTING ON THE CONDENSED FILE

If PF5 is pressed from the CONDENSE Function menu, CONDENSE links to the REPORT utility. The output condensed control file is supplied as the default primary control file, but can be overridden if, for example, the condensed data file is used as the secondary data file in the report. Note that the condensed data file fields are reported in alphabetical order with respect to the entire record, and that record type area boundaries are thus removed unless the default REPORT field sequence is modified. Report processing is described in Chapter 5, REPORT.

7.6 INVOKING A USER EXIT SUBROUTINE

The user can completely control the condensed file's contents through a User Exit subroutine. A User Exit subroutine can alter the contents of a record or omit a record entirely based on criteria more flexible than those supported by CONDENSE. The record type can be intercepted before it is placed in the buffer, and/or the entire buffer can be manipulated before it is written to the output file. Thus, a User Exit subroutine could, for example, enable a record type to be used in a sum field calculation but be omitted from the actual output file. A User Exit subroutine is particularly well suited for carrying out global modifications on the output condensed file as it is created, eliminating the need to perform repetitive modifications through DATENTRY. Invoking a User Exit subroutine involves the following steps, documented in the indicated sections.

- The file parameters of the User Exit subroutine must be supplied. Refer to Section 7.6.1 for details.
- A compiled or assembled subroutine must be constructed. Refer to Section 7.6.2 for details.
- The condensed file is created with the subroutine manipulating the file contents. Refer to Section 7.6.3 for details.

7.6.1 Parameter File Requirements

The file, library, and volume locations of existing subroutine object code must be supplied before creating the condensed file. The file parameters can be supplied on the Output Control File Specification screen, discussed in Section 7.2.3. The file parameters can also be supplied by modifying previous file definitions as described in Section 7.3.4.

7.6.2 Constructing the User Exit Subroutine

CONDENSE supports User Exit subroutines written in the VS Assembler, BASIC, COBOL, and FORTRAN languages. The subroutine must conform to the specific language requirements for an external subroutine as described in the VS Assembler Language Reference, VS BASIC Language Reference, VS COBOL Reference, or VS FORTRAN Reference.

CONDENSE passes control to the User Exit subroutine each time a record type contained in the parameter file is encountered and each time the buffer is written to the output file. Thus, the subroutine can manipulate the record type before it is placed in the buffer and/or the entire buffer contents before they are written to the output file. The subroutine can omit the record or record type from the file, or change the contents of the record or record type. CONDENSE and the User Exit subroutine communicate through two arguments, described as follows. These arguments must be specified in an argument list in the order in which they are described.

Argument	Length	Description
Record area	Record length of record type or buffer	The record type or buffer contents currently being processed by CONDENSE. The subroutine alters the condensed file's contents by changing the value of this argument. The field definition and record length in the subroutine should be consistent with those of the record type or buffer. In COBOL programs, the linkage section must contain an 01-level redefinition of this argument for each record type in the parameter file and for the output buffer. Field definitions are not required (but are permissible) unless the fields are referenced in the Procedure Division. A COBOL record description for each record type and the output buffer can be obtained from the CONTROL Copylib function. Subroutines in Assembler, BASIC, and FORTRAN are responsible for ensuring that the variable receiving the current record is dimensioned with a sufficiently large value. Note that BASIC programs must interconvert numeric data to the BASIC format for successful processing.
Record code	1 byte	A character value indicating the type of record passed. If CONDENSE passes a value of "I", the record is an input record type; a value of "O" indicates that an ouput buffer record is being processed. By changing the value of this argument to "B", the user can omit this record from the buffer and/or output file.

Examples of BASIC and COBOL subroutines follow. The BASIC subroutine examines output records and converts a last name of "JONES" to "ZEBRA". The COBOL subroutine specifies that output records containing fields with a last name of "JONES" or a full name of "HENRIETTA HARDY" should be bypassed (not written to the output data file). Note that the BASIC and COBOL subroutines operate on the same file.

BASIC Program

```
10  SUB "EXIT" ADDR (RECORD$, CODE$)
20  DIM RECORD$ 80, CODE$ 1
30  IF CODE$ = "I" THEN 50
40  IF STR(RECORD$,4,6) = "JONES " THEN STR(RECORD$,4,6) = "ZEBRA "
50  END
```

COBOL Program

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  USER1.

ENVIRONMENT DIVISION.
CONFIGURATION SECTION.
SOURCE-COMPUTER.  WANG-VS.
OBJECT-COMPUTER.  WANG-VS.

DATA DIVISION.
LINKAGE SECTION.

01  RECORD-AREA                  PIC X(80).

01  INQUIRY1 REDEFINES RECORD-AREA.
    03  ID1                      PIC  X(3).
    03  FNAME                    PIC  X(20).
    03  LNAME                    PIC  X(20).
    03  ADDRESS                  PIC  X(36).
    03  CODE 1                   PIC  X.

01  INQUIRY2 REDEFINES RECORD-AREA.
    03  ID                       PIC  X(3).
    03  NAME                     PIC  X(35).
    03  AGE                      PIC  X(3).
    03  JOB                      PIC  X(38).
    03  CODEX                    PIC  X.

01  INQUIRYO REDEFINES RECORD-AREA.
    03  ID-O                     PIC  X(3).
    03  LNAME-O                  PIC  X(20).
    03  CODE1-O                  PIC  X.
    03  ID-O                     PIC  X(3).
    03  NAME-O                   PIC  X(35).
    03  CODEX-O                  PIC  X.
    03  FILLER                   PIC  X(17).

01  RECORD-CODE                  PIC X.
    88 INPUT-REC                 VALUE "I".
    88 OUTPUT-REC                VALUE "O".
    88 RECORD-TO-BE-BYPASSED     VALUE "B".
```

```
PROCEDURE DIVISION USING RECORD-AREA
                         RECORD-CODE.
START-PROGRAM.
   IF INPUT-REC
   THEN EXIT PROGRAM.
   IF OUTPUT-REC
      PERFORM OUTPUT-RTN THRU OUTPUT-EXIT.
   EXIT PROGRAM.
OUTPUT-RTN.
   IF LNAME-O IS EQUAL TO "JONES" OR NAME-O IS EQUAL TO
        "HENRIETTA HARDY" MOVE "B" TO RECORD-CODE.
OUTPUT-EXIT.
   EXIT.
```

7.6.3 CONDENSE Processing with a User Exit Subroutine

When User Exit processing has been selected; CONDENSE automatically links to the specified subroutine through dynamic invocation of the LINKER utility. Thus, the user must not manually link the subroutine and CONDENSE object modules. When CONDENSE creates the output data file, it calls the subroutine each time a record type included in the parameter file is encountered and each time the output buffer is to be transferred to the output data file. The output data file reflects the specifications of both the parameter file and the User Exit subroutine.

7.7 A SAMPLE CONDENSE PROCEDURE

Although record type and field selection are necessarily interactive processes, workstation interaction can be minimized through a customized procedure. The parameter file location, CONDENSE function, input and output files, and User Exit subroutine can all be specified through a procedure. A complete list of CONDENSE GETPARMs can be found in Appendix A, File Management Utility GETPARMs; consult the VS Procedure Language Reference for details concerning Procedure syntax.

The following procedure runs the CONDENSE utility; supplies the file, library, and volume locations of the parameter file to be created; selects the parameter file creation function; names the User Exit subroutine; selects the condensed file creation function; supplies the location of the input data file and output control and data files; runs REPORT; and exits the CONDENSE Function menu and the CONDENSE utility. Note that processing within REPORT cannot be specified in this procedure since procedure values cannot be passed through a link.

```
PROCEDURE
ENTER PARMFILE PARMFILE=SEAN, PARMLIB=MLHCOND, PARMVOL=SYSTEM
ENTER FUNCTION 2
ENTER USEREXIT FILE=EXIT, LIBRARY=MLHOBJ, VOLUME=SYSTEM
ENTER FUNCTION 4
ENTER CONTROL CTLFILE=SINGLE, CTLVOL=SYSTEM
ENTER INPUT INFILE=MULTIPLE, INLIB=SLHLIB, INVOL=SYSTEM
ENTER OUTPUT OUTFILE=SINGLE, OUTLIB=SLHLIB, OUTVOL=SYSTEM
ENTER FUNCTION 5
ENTER FUNCTION 16
ENTER PARMFILE 16
RETURN
```

CHAPTER 8
A SAMPLE APPLICATION USING CONTROL, DATENTRY, AND REPORT

8.1 INTRODUCTION

This chapter presents a sample data management application that illustrates the combined use of three File Management Utilities: CONTROL, DATENTRY, and REPORT. This example demonstrates a payroll application for a hypothetical small business, and includes design and creation of a data file, entering data, and design and production of a weekly report based on the entered data. The data on the source forms is in the following format.

EMPLOYEE'S ID NUMBER	EMPLOYEE'S NAME	DEPT NUM.	REG. HOURS WORKED	OVERTIME WORKED	HOURS WORKED DURING WEEK	HOURLY WAGE
123456	Doe, John Q.	12	40	3	43	$5.00

With this information, a data entry and reporting system is designed and implemented. The three basic operations required follow.

1. Creation of the Control File
2. Creation of the Data File
3. Creation of a Report Definition File

8.2 CREATING THE CONTROL FILE

8.2.1 Planning the Data File

Since the control file defines the characteristics of the data file, it is important to decide on a data file layout before beginning the actual creation of the control file. (Careful planning of the data file before entering information on-line helps eliminate unnecessary control file modification.)

To a large extent, the layout of the data file is dependent on the design and requirements of the report to be produced. Therefore, the information to appear on the report must be considered when selecting and naming the fields in the data file. For this example, the report is to present a table of regular hours worked, overtime hours worked, total hours worked, wage rate, and total pay for each employee. The data is to be listed by department and organized alphabetically by employee name within each

department. Total pay is to be calculated for each department, as well as for the company. The report is to be run weekly, so only weekly information need be stored in the data file. Consequently, to produce this report the data file must contain such information as the department number, employee name, regular, overtime, and total hours worked, and wage rate.

Selecting the Fields

The first step is to determine which fields will comprise the data file. For this example, the existing organization and order of the data are maintained, but the field, "Employee's Name", is expanded into two separate fields: "Last Name" and "First Name/Middle Initial." Additionally, a cumulative field for "Total Hours" is created, representing the sums of the values entered into the "Regular Hours" and "Overtime Hours" fields. A field name of eight or fewer characters is then assigned to each field. These fields are appropriate, both in terms of their compatibility with the original data and of the information required by the report.

Since design of the data file is most easily and clearly accomplished by developing a table of field information, all data file specifications are tabularized as they are selected. The data file thus far has the following fields.

Old Field Name	New Field Name
Employee's Number	EMPNUM
Employee's Name	LNAME
	FNAME
Department	DEPT
Regular Hours	REGHOURS
Overtime Hours	OVRHOURS
Hourly Wage	WAGERATE
Total Hours	TOTHOURS

Specifying Internal Format

Once the data fields are selected and named, an internal format is specified for each field. However, before selecting the internal formats, it is important to understand the distinction between external and internal formats. The external format is the way in which a field's data appears to the user and is of two types: alphanumeric and numeric. The internal format specifies the mannner in which a field's data is stored internally in the computer. A field's internal format is transparent to the user. There are five types of internal format, four of which are used in this example: character (C) for all alphanumeric fields, zoned (Z), unsigned (U), and packed (P) for numeric fields. Binary format is not discussed here. (Refer to Chapter 3, CONTROL.)

Since all alphanumeric fields must have an internal format of C, they share the same type of internal representation (i.e., each alphanumeric character requires one byte of storage). For example, ABC1 is represented internally in 4 bytes.

There are three ways in which numeric data will be represented internally in this example. These three internal formats -- zoned, unsigned, and packed -- differ in the number of bytes they require to represent a given piece of data. Zoned format requires a full byte for each digit, exclusive of sign and decimal point. For example, "S1234" (S = sign of + or - 1234) requires four bytes of storage in zoned format. The sign is stored in the high order bits of the last byte (see below). Unsigned format also stores one digit per byte, but does not store signs (if a sign is added via the data edit options, the sign is stored in a full byte by itself). (Note that unsigned format (U) cannot have validation tables and cannot be an accumulator field.) Conversely, packed format requires only a half-byte of storage per digit, plus one additional half byte for the sign of the data. Therefore, packed format requires three bytes of storage to represent "S1234".

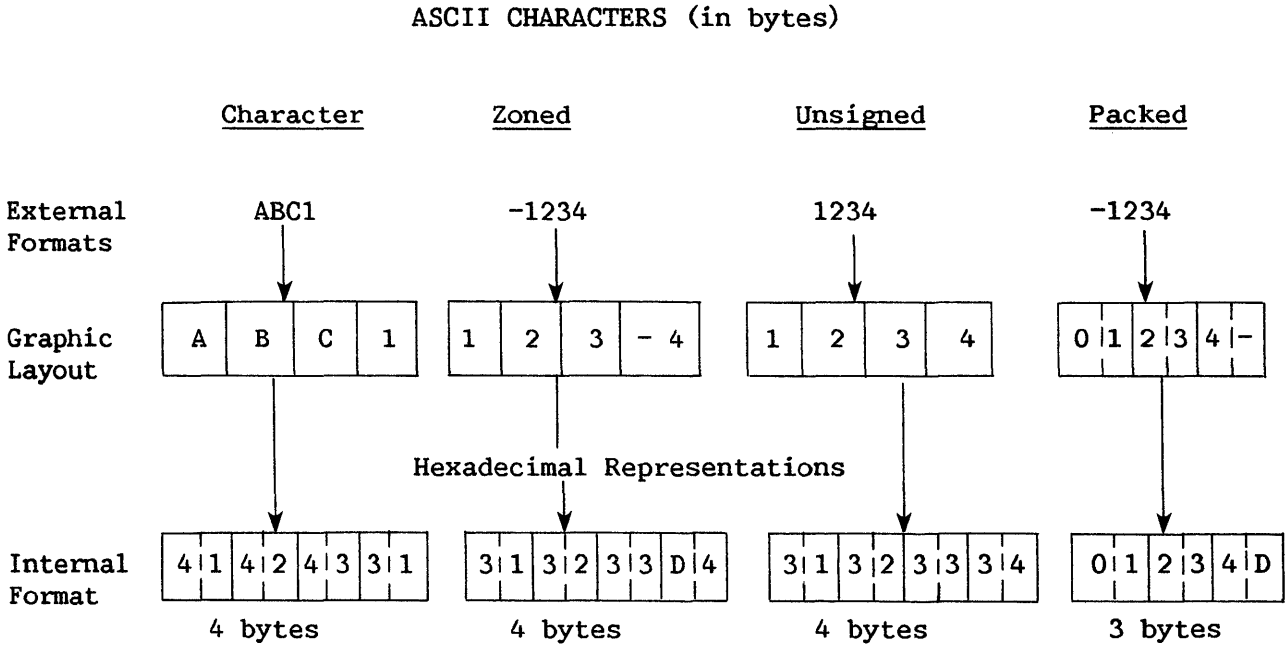


Figure 8-1. Data Representation

The format for numeric fields is important because it can affect overall file space requirements and/or CPU time, as well as screen representation of the field length. Choice of format should be dependent on field size and usage. Packed format requires less storage space than zoned or unsigned formats for fields exceeding two bytes in length. Similarly, packed format requires less CPU time than zoned format for fields used in arithmetic operations. Less CPU time is needed because all arithmetic operations are performed in packed format internally, necessitating internal conversion of zoned and unsigned fields to and from packed format. For these reasons, it is preferable to select numeric format based on the parameters specified on the following page.

Format	Suggested Selection Criteria
Z	Signed numeric fields of 1 or 2 digits on which no arithmetic operations will be performed.
U	Unsigned numeric fields on which no arithmetic operations will be performed.
P	Signed numeric fields of 3 or more digits, as well as those on which arithmetic operations will be performed.

The File Management Utilities restrict the sizes for all three formats to no more than 15 digits without a decimal point and 14 digits with one.

Using these parameters, the data fields are assigned internal formats as shown below. Note that the field EMPNUM is designated an unsigned (U) field. This is because it is always unsigned and so does not require the extra pseudoblank automatically provided by packed format. This extra pseudoblank is eliminated by using the unsigned format, reducing possible data entry confusion, although it does extend the data record by two bytes.

Field Name	Data Format
EMPNUM	U
LNAME	C
FNAME	C
DEPT	U
REGHOURS	Z
OVRHOURS	Z
WAGERATE	P
TOTHOURS	P

Specifying Field Lengths

The number of characters or digits and the internal field length must be specified for each field. The number of characters or digits in each field is a working value from which the internal field length is determined. The number of characters or digits is selected through examination of the data; values for the three HOURS fields, as well as the DEPT field, range from 1 - 99, and thus require two digits. Similarly, WAGERATE includes values like $2.75, and therefore requires three digits. (The dollar sign and decimal point are edit characters that can be added when the Report Definition file is created.)

Using the internal formats and the number of characters or digits, the internal field lengths then are determined from the algorithms given below. (The internal field length is the number of bytes required to represent the specified number of characters or digits on the data file record. The calculations that follow, as well as those for external length, make an allowance for the required place for a sign in both zoned and packed.) The sum of the internal field lengths is the total record length of the data file.

Internal Format	Internal Field Length
C	Number of Char. or Digits required
Z	Number of Digits required
U	Number of Digits required
P (even number of digits)	(Number of Digits/2) + 1
P (odd number of digits)	(Number of Digits/2) + .5

It is not necessary to enter the external field length when creating a control file, since the external length is calculated automatically from the internal length using the algorithms shown below. (The external field length is the number of pseudoblanks appearing on the data entry screen.) However, since some applications require adjustment of the external field length, it is useful to be familiar with these calculation algorithms.

Internal Format	External Length
C	Internal length
Z (decimal positions = 0)	Internal length + 1
Z (decimal positions ≠ 0)	Internal length + 2
P (decimal positions = 0)	(2 x Internal length)
P (decimal positions ≠ 0)	(2 x Internal length) + 1

The data file information, including the internally calculated values for external length, now appears as follows.

Source Field Name	Field Name	Data Format	Number of Char.	Internal Length	# of Dec. Positions	External Length
Employee's Number	EMPNUM	U	6	6	0	6
Employee's Name	LNAME	C	20	20	--	20
	FNAME	C	10	10	--	10
Department	DEPT	U	2	2	0	2
Regular Hours	REGHOURS	Z	2	2	0	3
Overtime Hours	OVRHOURS	Z	2	2	0	3
Hourly Wage	WAGERATE	P	3	2	2	5
Total Hours	TOTHOURS	P	3	2	0	4

Note that all of the external length values listed are those that are internally calculated by the system; hence, the user need not enter an external length value for any of the data fields when creating the control file.

Specifying the File Organization

Following the specification of internal field formats, the data file record type must be selected. The record type selected is based on whether or not the file is to be indexed and whether or not the data is to be modified. Note that variable-length records cannot be used for a consecutive file that requires modification. For the purposes of this example, the data file has fixed-length records.

Next, the organization of the file is selected as either indexed or consecutive. For the purposes of this example, the data file contains one record for each employee, and records are updated periodically. Since it is useful to access data records through particular fields, the file is an indexed file.

For this example, it is convenient to access records by employee number, employee name, and department number. Since each employee has a unique employee number, EMPNUM is designated as the primary key. LNAME, FNAME, and DEPT are selected as alternate keys and can, therefore, contain duplicate values.

Specifying Validation Methods

Next, any data validation is specified by selecting tables, ranges, or user exit routines. Since the hypothetical company is a small one, a table look-up is specified for DEPT to verify all of the values entered into this field. Additionally, a range of values is specified for WAGERATE to ensure that payments to employees fall within the accepted limits. Validation parameters are chosen, based on examination of the data; WAGERATE is given a $2.95 to $9.00 an hour range. These validation methods are summarized with their table values and ranges below.

Additional Characteristics

Finally, any characteristics peculiar to a field are noted. For example, a display code of "1" for DEPT causes the present department number to remain displayed on each data entry until modified. Two decimal positions and a dollar sign are inserted for WAGERATE, the cumulative field TOTHOURS for the fields REGHOURS and OVRHOURS, and UPDATE = 1 is specified for TOTHOURS, to prevent TOTHOURS from appearing on the data entry screen. (Since TOTHOURS sums the values entered into REGHOURS and OVRHOURS, it should not be modifiable from the data entry screen.) The completed table now appears as follows.

Source Form Field Name	Index Keys	Utility Field Name	Int. Format	Num. Char	Int. Len.	Validation Method	Other Features
Emp. Num.	P	EMPNUM	U	6	6		
Emp. Name	A1	LNAME	C	20	20		
	A2	FNAME	C	10	10		
Department	A3	DEPT	U	2	2	Tab: DEPT	Display= 1
Regular Hours		REGHOURS	Z	2	2		CumFld=TOTHOURS
Overtime Hours		OVRHOURS	Z	2	2		CumFld=TOTHOURS
Hourly Wage		WAGERATE	P	2	2	Range: 2.95- 9.00	2 dec.pos. Doll = 2
Total Hours		TOTHOURS	P	3	2		Update = 1

Totals: 8 Fields 46 byte data record

Now that all of the fields are defined, the data is organized into a tabular form that facilitates creation of a control file. As shown below, columns are swapped to match the order of the specifications in the control file definitions screens, and the fields are placed in the order in which they are to be entered into CONTROL. TOTHOURS is placed before REGHOURS and

OVRHOURS, since an accumulator field must be defined before its source fields. Also, the data formats and internal lengths are checked to insure no mistakes were made. Such checking procedures help to eliminate control file modifications later. The control file may now be created using this table.

Field Name	Index Keys	Int. Format	Int. Length	Ext. Length	Other Features	Validation Methods
EMPNUM	P	U	6	System		
LNAME	A1	C	20	System		
FNAME	A2	C	10	System		
DEPT	A3	U	2	System	Display = 1	Table: DEPT
TOTHOURS		P	2	System	Update = 1	
REGHOURS		Z	2	System	Cum Fld = TOTHOURS	
OVRHOURS		Z	2	System	Cum Fld = TOTHOURS	
WAGERATE		P	2	System	2 dec. pos. doll/com = 2	Range: 2.95-9.00
Totals:	8 Fields		46 byte data record			

8.2.2 Entering the File and Field Information

Information for a control file is entered by running CONTROL. File, library, and volume names must be specified. Next, the creation option is selected from the Control File Options menu (PF2, Create A Control File). The subsequent screen requests data file header information. Using the prepared table of field information, the necessary parameters are specified on the screen as follows.

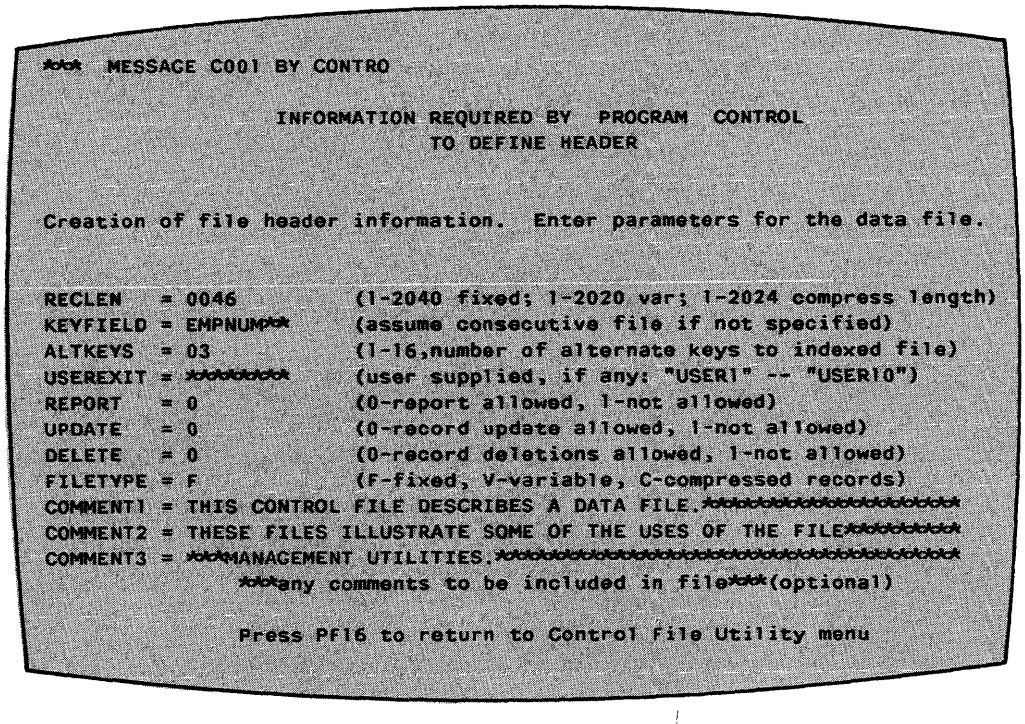


Figure 8-2. Control File Header Screen

These values indicate the data file record length is 46 (as computed), the primary key is EMPNUM, and there are three alternate keys. Additionally, there is to be no user exit routine; the data file can be reported upon, updated, and have deletions performed on its data, and the file is designated as having fixed-length records. (Note that the parameters indicating reporting and updating of the data file govern access only through the File Management Utilities; access through other methods is not controlled.) Finally, a brief description of the data file is included in the comment records.

NOTE

There are four types of records in the control file: a header record, an alternate key record, comment records, and data field records. The header record contains all of the identifying information concerning the data file, and consists of such fields as RECLEN and KEYFIELD. The alternate key record contains the names of and whether or not duplicate values may be used for alternate keys. The comment records contain user comments about the file that appear when the control file is listed. A data field record contains all of the information about one data field, such as EMPNUM, and consists of such fields as FIELD NAME, INTERNAL FORMAT, and INTERNAL LENGTH. In the data file, a record is a set of related data values, with one value for each specified field. In an indexed data file, a record is the set of data for a given key field value, such as all of the data in the fields LNAME, FNAME, etc., for a given EMPNUM value. The record types of the control file are illustrated in Figure 8-3. Refer to Appendix C, Control File Record Formats, for more information about the contents of a control file.

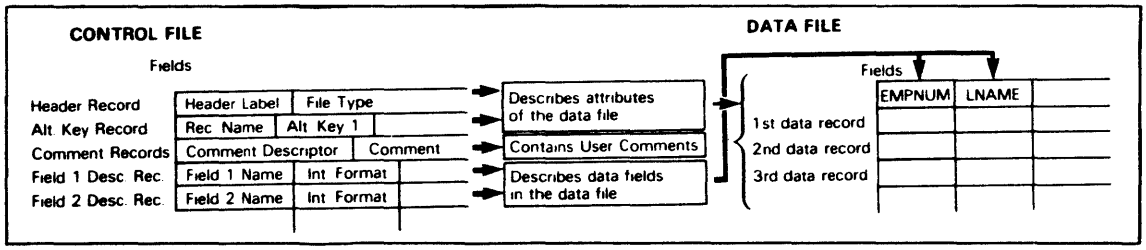


Figure 8-3. Control File Record Types

The subsequent screen requests the names of the three alternate keys and whether or not they may contain duplicate values. LNAME, FNAME, and DEPT are entered in order, and all may have duplicate values. (The response to the Duplicates Allowed option is YES.)

The following screen prompts for the specifications of the first field. Specifications of the first field, EMPNUM, are entered as shown. The starting position for EMPNUM is entered as "1" and will be computed internally for all subsequent fields. The external length values not entered for any of the fields are computed by the system. Note that changes to defaults are maintained for the definition of subsequent fields; therefore, each field entry must be carefully checked. (For example, if the internal format is set to P for one field, it remains P for all following fields unless it is changed.)

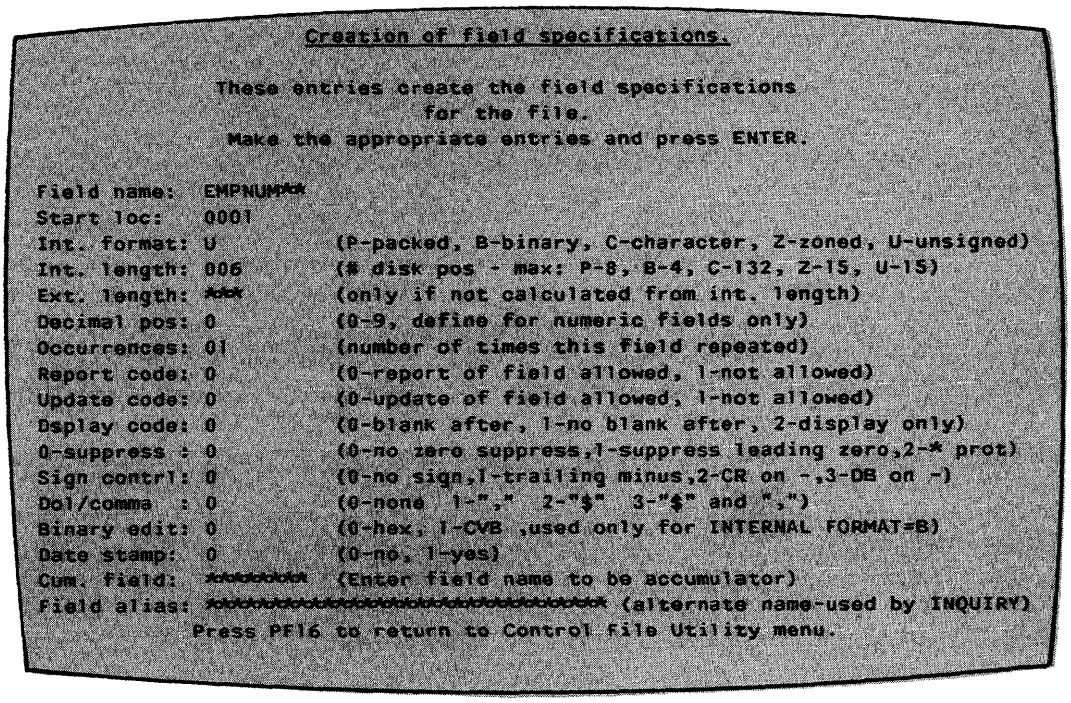


Figure 8-4. Field Specification Screen

A Table-Range Specification screen is displayed following the definition of all fields 16 or fewer bytes in length. For example, following completion of the DEPT field definition, the table DEPT is specified. Table data can be entered following table specification or at a later time; for the purpose of this example, it is assumed that all data is entered following table specification.

Assume that only DEPT values of 8, 12, and 15 are to be included in the data file. They are entered in the table displayed after DEPT has been defined. (Note: When all table values have been entered, press PF3 to continue field definitions. The alternative, PF16, deletes all record of the field and its table.)

Once all the fields, tables, and ranges have been specified, creation of the control file is completed. The newly created control file can be examined by pressing PF6 from the Control Options menu. A data file can now be created, or the control file can be altered using the methods discussed in Chapter 3, CONTROL.

8.3 CREATING THE DATA FILE

The DATENTRY utility may be accessed either from the Control Options screen (PF9) or from the Command Processor menu. The first step in creating a data file is to specify its name and location and the name and location of its corresponding control file. Since it is recommended that data files be given the same name as their control files, this data file is named PAYROLL and is located in library IDDATA. Following this, the creation option is selected from the DATENTRY Options menu (PF2, Create a Data File). The subsequent screen (INDFILE) requests the storage medium for the data file (DISK or TAPE), the retention period and file class of the data file, and the approximate number of records that constitute the data file.

Once the parameters have been specified, the screen for data entry is displayed. Field names are presented in the order in which their pseudoblanks are to be filled. Data is entered into the appropriate fields from the source form shown below, with tabbing to fields and ENTER keyed after the completion of each record. Note that a comma is included after each value for LNAME, and is treated as a part of the value entered. Any data that does not conform to the range or validation criteria established for its field is rejected, and an appropriate error message is displayed.

EMPNUM	LNAME	FNAME	DEPT	REGHOURS	OVRHOURS	WAGERATE
050025	Roberts,	Anne R.	08	40	5	$ 5.25
200001	Jones,	Fred A.	08	40	0	$ 5.00
200101	Adams,	Susan B.	08	40	8	$ 4.75
060057	Faulkner,	Cindy	12	40	3	$ 5.50
100001	Smith,	Thomas D.	12	35	0	$ 5.00
100025	Kennerly,	Sarah F.	12	40	0	$ 3.50
400101	Harrigan,	Maureen	15	40	0	$ 5.50
326001	Butler,	Jean K.	15	40	15	$ 6.00
140001	Invalid,	I.M.	10	25	0	$ 6.00

The first record to be entered into the data file appears as shown in Figure 8-5.

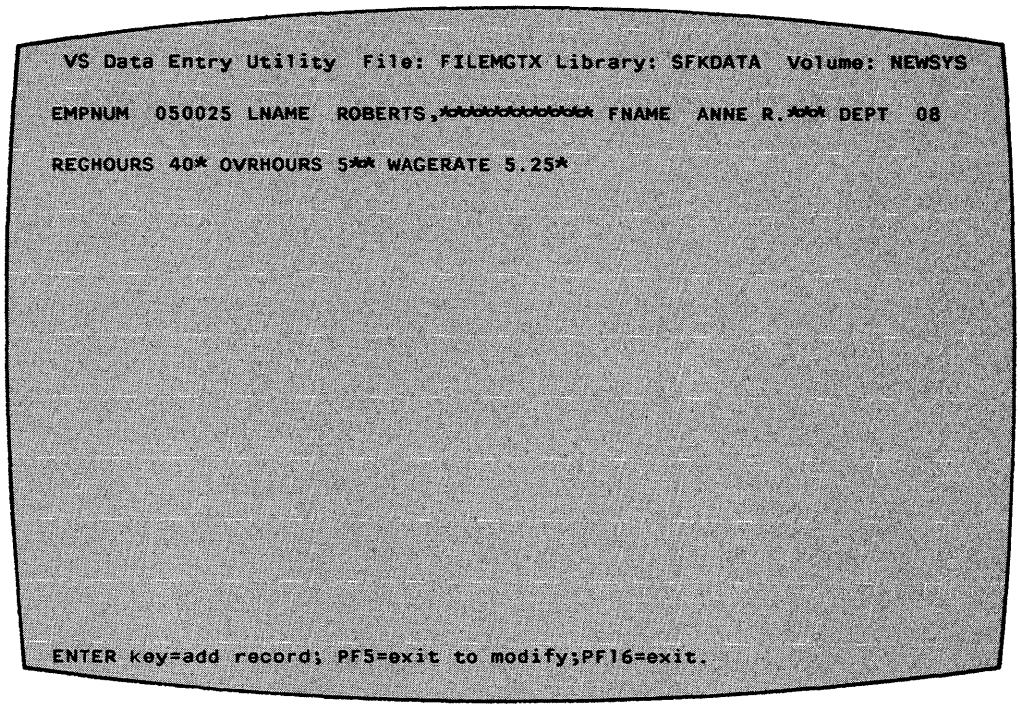


Figure 8-5. Data Entry Screen

Note that the last record to be entered contained invalid data, and so could not be entered.

Once this set of data has been created, a report can be run on it, or it can be modified to simulate successive weeks of REGHOURS and OVRHOURS data. Note that any change made in the values of the accumulator source fields results in automatic changes in the values of the accumulator field (i.e., the values for TOTHOURS are recalculated each week). The data records are accessed by any of the key paths specified for the data file and, within a key path, by sequence of records or key value.

Once all of the data has been entered, PF16 permits exit from DATENTRY back to the DATENTRY Options menu, and from there to either the CONTROL Options menu, or the Command Processor menu. A report can now be designed and printed.

8.4 CREATING THE REPORT DEFINITION FILE

8.4.1 Introduction

REPORT permits the user to design a report to be run on a data file and a control file (or two such sets of files). REPORT places the specifications for a report into a report definition file (RDF), which then may be used repeatedly to produce reports on the specified data file(s).

8.4.2 Planning the Report Format

As in control file creation, the first step in creating a report definition file is to plan the report format before entering the report specifications. To design this report, it is useful first to sketch out a rough draft of the report format. This draft is then used as a basis for a more detailed design. The draft should not be created with the details of the report in mind; rather, it should reflect the desired final format of the report. The structure of the report is then created from the draft.

As discussed earlier, this report presents the regular, overtime, total hours worked, wage rate, and total pay for each employee. This data is to be organized alphabetically by employee name (last, first) within departments, and by departments in ascending numeric order. Total wages are to be calculated and printed in summary lines for each department, as well as for the company.

Designing a Draft

When designing the draft format, it is easiest to begin with the columns. These columns, as discussed earlier, are department number, employee name (last, first), regular hours, overtime hours, total hours, wage rate, and total pay. With this information, column headings may be assigned (DEPT, EMPLOYEE, REGULAR HOURS, OVERTIME, TOTAL HOURS, WAGE RATE, and TOTAL PAY). Next, a one-line report title is selected (EMPLOYEE WAGES). Finally, a descriptor for each summary line is specified (TOTAL WAGES:). The draft, with x's representing data, appears as follows.

```
                                EMPLOYEE WAGES

DEPT  EMPLOYEE     REGULAR HOURS  OVERTIME  TOTAL HOURS  WAGE RATE    TOTAL PAY

XX    XXXXXXXXXX   XXXXXXXX       XXXXXX    XXXXXXX      XXXXXXX      XXXXXXX
XX    XXXXXXXXXX   XXXXXXXX       XXXXXX    XXXXXXX      XXXXXXX      XXXXXXX
                                                      TOTAL WAGES:    XXXXXXXX
```

Figure 8-6. Design of a Report

Using this draft, the structure and details of the report can now be specified. Fields must be selected, organized, and designated for specific uses within the report (such as new fields, sort fields, and control break fields). Titles and headings must be assigned to specific report attributes (such as control break descriptors). Furthermore, appearance attributes must be specified, such as spacing between fields and between sections. Once these design decisions are made, the report specifications can be entered via the REPORT utility.

Selecting Fields

The fields needed for the printing of a report are selected through examination of the columns on the rough draft. The fields required for this report are DEPT, LNAME, FNAME, REGHOURS, OVRHOURS, TOTHOURS, and WAGERATE. Note that the draft specifies a column for total pay to date, yet no such data

field exists. Thus, a new field must be created for this report. This field is named TOTPAY, is numeric, and is the sum of REGHOURS multiplied by WAGERATE, plus OVRHOURS multiplied by 1.5 times WAGERATE. (Overtime pay is time and a half.) Additionally, since TOTPAY is a dollars field possibly running into four figures, it will have two decimal places, a dollar sign, and a comma, giving it a total external length of eight. REPORT automatically adds two places to this length: one for the decimal point and one for the sign, giving a total final external length of 10. This external length is discussed more fully below.

Once the fields have been selected, their sequence on the report can be specified. Field sequence introduces another issue, the selection of fields for the report. Fields used in preparing for a report, but not printed in it (such as EMPNUM in this example), are assigned a sequence number of 99. A sequence number of 99 causes a field to be accessible, but not printed in the report. This option also provides the user with the flexibility of using these fields in the report at a later time.

The sequence numbers for all of the fields in the data file PAYROLL can now be assigned. The order of fields is determined through examination of the data columns from left to right. All unused fields (such as EMPNUM) are given a sequence number of 99, and, therefore, do not appear on the final report. The sequence of the fields is as follows.

Field	Seq. No.	Field	Seq. No.
EMPNUM	99	REGHOURS	04
LNAME	02	OVRHOURS	05
FNAME	03	WAGERATE	07
DEPT	01	TOTPAY	08
TOTHOURS	06		

Next, the fields on which the data is to be sorted can be specified. Since the data on the report is to be organized alphabetically by employee name (last, first) within each department, and by department in ascending order, the data file must be sorted on these fields. Therefore, sorting is performed in ascending order, first on DEPT, then LNAME, and finally on FNAME.

Finally, any control field to be used in this report must be selected. A control field is a field for which a control break is taken every time the value of the control field changes. A control break causes a summary line to be printed (if summary lines are specified) and blank lines to separate one control section from another (if blank lines are specified). Since this report is to print a summary line (total pay-to-date) for each department, both a control field and summary lines must be specified.

Since a summary line is to be printed each time the value of DEPT changes (i.e., for each department), DEPT is the control field. Note that there are no other control fields, since no further subdivision of the data is required. At this time, it is useful to determine how many blank lines are to separate control break sections, since this option is specified in REPORT at the same time as the control field.

Summary Specifications

Once the control field has been selected, it is useful to specify the details of the summary lines. The available summary options are total, maximum, minimum, and average values for all numeric fields used in the printing of the report (here, REGHOURS, OVRHOURS, TOTHOURS, WAGERATE, and TOTPAY). These summary values are computed at each control break and at the end of the report. For this particular report, the only desired summary information is the total value for TOTPAY. Therefore, this is the only summary option specified. This total value is printed for each department, as well as at the end of the report.

Selecting Titles and Headings

Following selection and organization of the fields to be used in the report, titles and headings can be specified. The main title for reports can be up to three lines long and sixty characters wide. The title for this report is "EMPLOYEE WAGES", on one line, as specified in the rough draft. This title is automatically centered on the top of the report by the Report utility.

The column headings for the selected columns are as specified in the rough draft. Since there is only one column heading for the two columns for LNAME and FNAME, the field FNAME is not given a column heading. To center the one heading over the two fields, the word EMPLOYEE is broken into two parts, the first part specified as the column heading for LNAME, and the second part specified as the column heading for FNAME. This is described in detail in the Selecting Spacing section below.

Finally, the rough draft specifies that "TOTAL WAGES:" is to be printed on each control break summary line, just before the total pay-to-date figure. "TOTAL WAGES:" is, therefore, the control break descriptor. (A control break descriptor is a user message printed at the specified position on the summary line of each control break.) The starting column number for "TOTAL WAGES:" will be selected at the same time as the spacing for the data columns. Note that the level of the control break descriptor must correspond to the level of the control field. In this case, both are level 1.

Selecting Spacing

Placement of column headings and data via spacing is important in an attractive report. Spacing is accomplished through careful examination of both the data and the width of the paper on which the report is to be printed (the print line length).

The first consideration in report spacing is whether or not all of the data and headings can fit onto the selected paper. Since this report is not to exceed 80 characters in width (the width of this page), the data field lengths and column heading lengths chosen, as well as the spaces between fields, must not exceed 80 characters. To ensure this, it is easiest to lay out the column headings and data fields with their corresponding print lengths. Note that the user can use the Print Headings and Dummy Detail Lines option of REPORT to view a simulated report in draft format. X's and 9's are used to represent data, and the total width for a given column is its heading

or field length, whichever is larger. (For example, the department column heading is 4 characters long, while its data is only two; therefore, the column is 4 characters wide.) The report might be laid out as follows.

```
DEPT  EMPLOYEE          REGULAR HOURS  OVERTIME  TOTAL HOURS  WAGE RATE TOTAL PAY
  | |                 | |            | |       | |          | |       | |
99| |XXXX...XXXXXXX| |99          | |99     | |999       | |$99.99 | |$9,999.99
  | |                 | |            | |       | |          | |       | |
 4+|2| +   31    +   |2| +  13   +  |2| + 8 + |2| +  11  + |2|   + 9 +|2|   + 9 =97
```

Total print line length: 97 (85 characters, 12 spaces)

Figure 8-7. Spacing in a Report

Clearly, the print line length poses a problem for this report. It must reduced by 17 places before report definition can continue. Print line reduction is most easily accomplished in this case by shortening fields or column headings that are longer than necessary, rather than by eliminating fields.

To eliminate excess space within fields, the actual data can be compared to the allotted external field lengths to determine if the spaces are, in fact, occupied by data. If any external length is found to be unnecessarily large, it may be truncated on the report to eliminate wasted space, and to provide the user with greater flexibility in positioning the columns. For example, if there are no last names 20 characters long, there will be excess spaces between all of the last and first names of employees. These spaces can cause the report print line length to exceed 80 characters. They also make the report appear discontinuous and unattractive, and prevent the user from positioning the columns close together.

Examination of the data fields shows both name fields have larger external lengths than are required by their data. The longest entries for both LNAME and FNAME are only 9 characters, so the total space allotted to EMPLOYEE need only be 19 spaces long (includes a space between last and first names). This is a print line reduction of 12 places from the 31 originally calculated for these fields. In addition to reducing the oversized print line, the truncation of the external lengths of the fields LNAME and FNAME creates a more attractive report by allowing the first and last names of employees to be printed closer together.

Despite this print line length reduction, the print line is still five columns too long. This is most easily remedied by reducing a large column heading. In this case, the logical choice is to reduce the heading REGULAR HOURS to HOURS. This permits all of the columns to remain on the report.

The number of characters on the print line has now been reduced to 65, leaving 15 places for spaces between the fields. These spaces are used to position the columns (composed of headings and data). First, the spacing between columns is assigned, and then placement of headings and data within each column is determined.

In assigning the spaces between the columns, note that there are six places where spaces are to be inserted to separate the columns. (The column for DEPT is placed at the left margin, and, therefore, has no spaces preceding it.) The columns for LNAME and FNAME have no spaces separating them, for a reason to be explained later. Since it is appropriate to widely separate the TOTPAY column from the other columns, the spaces before the last column should be greater than between the other columns. The 15 spaces now can be divided among the columns. The fields LNAME, REGHOURS, OVRHOURS, TOTHOURS, and WAGERATE are to have 2 spaces before each of them, and the TOTPAY column is to be preceded by 5 spaces, thus separating it from the main body of the report. The report now appears as follows.

```
DEPT  EMPLOYEE              HOURS  OVERTIME  TOTAL HOURS  WAGE RATE     TOTAL PAY
    | |                    | |    | |       | |          | |         | |
99  | |XXXXXXXXX XXXXXXXXX| |99   | |99     | |999       | |$99.99   | |$9,999.99
    | |                    | |    | |       | |          | |         | |
 4  |2|   9      1     9   |2| 5  |2|   8   |2|    11    |2|    9    | 5|    9
```

Figure 8-8. Total Spaces in a Report

Next, the column headings are centered within their respective columns. This is done column by column. Column headings are positioned by the user; data is automatically centered under its respective column heading by REPORT. Since the column heading DEPT is larger than its corresponding two digit data, the data is to be centered under its heading with a space before and after it. Similarly, the two-digit data for both REGHOURS and OVRHOURS is to be preceded by three spaces, and followed by two and three spaces, respectively. The three-digit data for TOTHOURS is to be preceded and followed by four spaces within its eleven-space column.

The spacing for the WAGERATE data is handled in a similar fashion. Since its data length is smaller than its 9-character heading, the WAGERATE data is preceded and followed by two spaces. (Note that because REPORT adds one place to the original external length of four to allow space for the dollar sign specified in the control file, the data has an external length of six.)

The calculations for TOTPAY are somewhat more involved than for the other fields. The external length for the new field TOTPAY is originally specified as eight to allow space for its six digits, dollar sign, and comma. REPORT automatically adds two places to give a total external length of ten: one place for a decimal point and one for the sign. However, since a sign is not appropriate for this report, it is to be deleted when the report information is entered. Therefore, the external length of TOTPAY is reduced to nine. Since the heading TOTAL PAY is the same length as its data, no centering of either heading or data is required.

Note that there is an alternative to reducing the external length of TOTPAY. The length could be retained as 10, since the zero-suppression that will be specified for the field eliminates unnecessary places. However, a data length of 10 exceeds the title length of 9, causing the data to extend one place to the right of the title. Since the data would most likely never require the full 10 places, it would not align with the left end of the

heading either, giving the column an unbalanced appearance. This could be rectified by indenting the title one space (i.e., "xTOTAL PAY", where x = space), giving the column an overall width of 10.

Positioning the heading for the LNAME and FNAME columns introduces a conflict: since the column is to have only one heading, EMPLOYEE, the heading must be centered over both fields. This is accomplished by entering part of the heading as a column heading for each field. To create a continuous single heading, no spaces between these two fields can be specified, since spaces would cause EMPLOYEE to be printed in two parts. However, it is desirable to print a space between the comma (end) of the longest LNAME entry and the beginning of its corresponding FNAME entry. This spacing is accomplished by assigning LNAME an external length of ten instead of nine, thus causing a space to be printed after the longest LNAME entry. (Note that all other LNAME entries are shorter than 9 characters, so they are already separated from their FNAME entries.)

To position EMPLOYEE, the center of the column must first be determined. Since EMPLOYEE is 8 characters long and the data column is 19, EMPLOYEE is preceded by 5 spaces and followed by six. Thus the column heading for LNAME is "xxxxxEMPLO", and the column heading for FNAME is "YEExxxxxx", where x = space. In summary, therefore, the total print line length for both name fields remains 19, a single title is achieved, and all of the data is separated by at least one space.

Finally, the control break descriptor must be positioned within its summary line. Since the last column begins at print line position 71 and the descriptor is twelve characters long, "TOTAL WAGES:" begins at print line position 59.

8.4.3 Specifying the Report

Once the design of the report is accomplished, the report specifications can be entered. The REPORT utility is accessed through either PF10 from the CONTROL Options menu or from the Command Processor menu. Specification of the report begins with selecting PF2, Create a Report Definition, from the REPORT Main menu.

Specifying the Source Files

The first screen requests the ID of the report being created. To maintain consistency with the control and data files, the report is named PAYROLL. Additionally, this screen requests whether an additional data file is to be used. Since only one data file is used for this report, the default value, NO, is retained.

The next screen requests the name and location of the primary data file. A second screen requesting the same information for the corresponding control file can then be displayed, if the control file cannot be located from system and user defaults.

Field Selection

Once the source files have been specified, the fields to be used in the report are selected (first from the primary data file and then from any secondary) . The fields in the data file are presented to the user, who selects individual fields by placing an "X" next to each chosen field. As discussed earlier, all of the fields in the data file PAYROLL are selected for this report. An "X" is typed next to each field name, and ENTER is pressed.

New Field Specification

The next screen requests whether or not new fields are to be defined. Since TOTPAY is a new field, YES is entered. The subsequent screen requests the names of new fields, whether they are numeric or character fields, what their maximum lengths are, and, if numeric, how many decimal positions they have. Since TOTPAY represents a dollar value, it is an eight-byte numeric field containing two decimal positions. It is shown as follows.

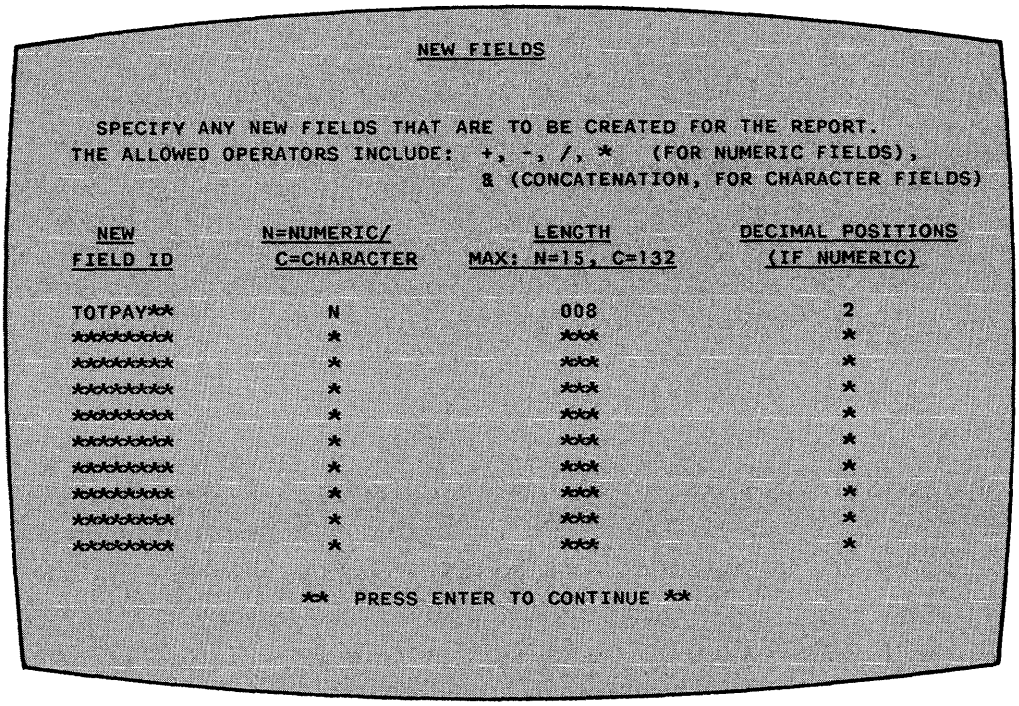


Figure 8-9. New Fields Specification Screen

The following screen requests the origin of the new field TOTPAY. New fields can be created from combinations of existing fields, constants, and/or literals. Numeric fields and/or constants can be added, multiplied, divided, or subtracted with each other. Character fields can be concatenated with each other or with literals. As discussed earlier, TOTPAY may be expressed as 1.5 *OVRHOURS + REGHOURS * WAGERATE. Therefore, this information is entered as shown on the following page.

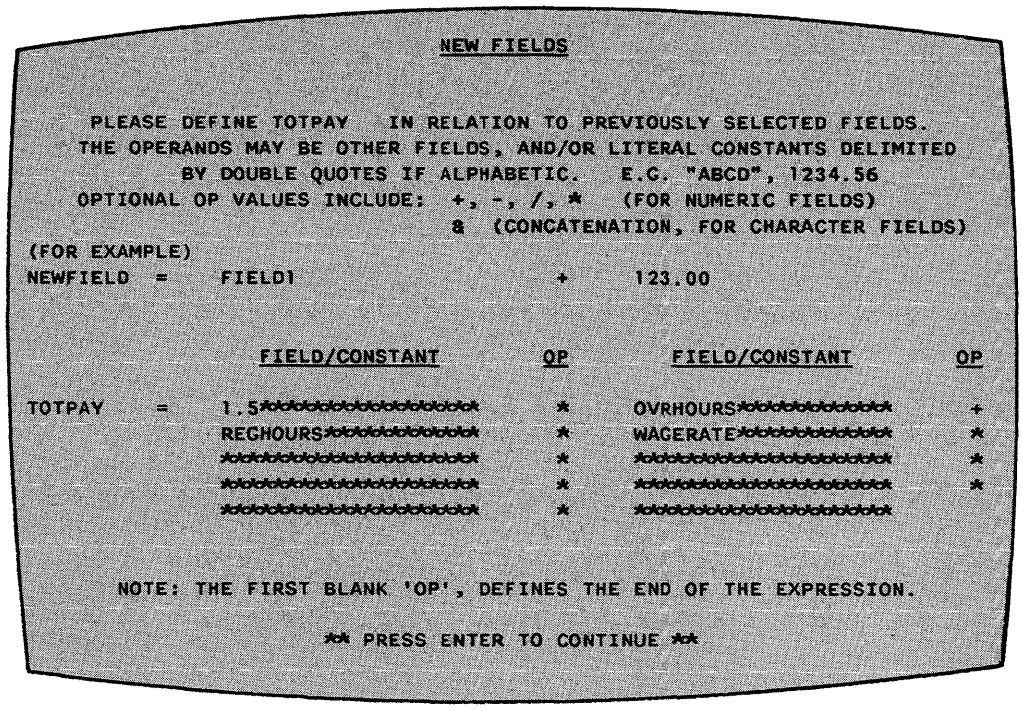


Figure 8-10. New Field Definition Screen

Once the fields have been selected and the new fields defined, the format of the report is specified through the options on the Report Definition Options menu. These options can be accessed individually from the PF keys, or in sequence, as in this example, by keying ENTER after each screen is completed.

Titles and Headings

The first Report Definition screen requests the main title of the report, the page headings (which appear on all but the first page of the report) and the control break descriptors. The report title, as discussed earlier, is entered as EMPLOYEE WAGES. No page heading is specified, since this report most likely is too short to require a second page. Finally, the level 1 control break descriptor is specified as TOTAL WAGES:, beginning in column 59.

Column Headings

The subsequent screen requests column headings and subheadings. Since all of the data fields were selected for this report and the unnecessary fields have not yet been removed by means of a sequence number of 99 (such as EMPNUM), all of the data fields are displayed in alphabetical order. The column headings (there are no subheadings) are entered, with the default field name retained for the field not to be printed in this report (i.e., EMPNUM).

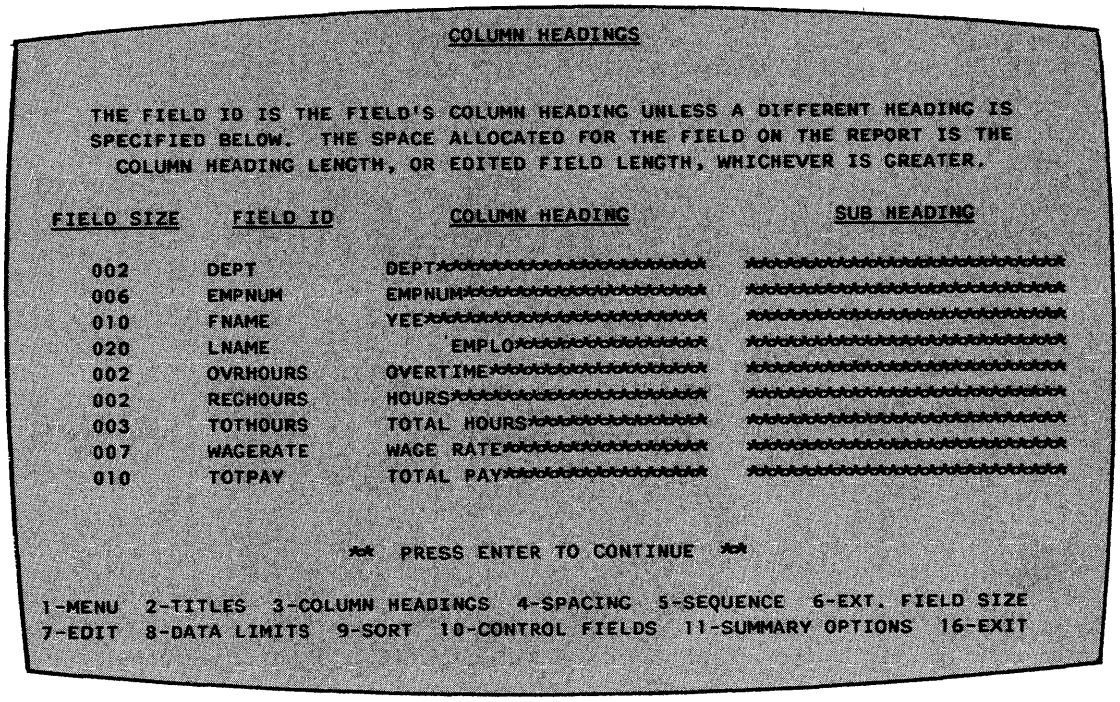


Figure 8-11. Column Headings Screen

Spaces Between Fields

The next screen requests the number of spaces before each field on the report. The displayed default value is two and is assigned to each field. Again, the fields not to be printed are displayed since they have not yet been assigned sequence numbers of 99. These fields should be allotted no spaces (i.e., 0) before fields. The spaces before fields are entered as determined earlier, and shown as follows.

FIELD ID	SPACES	FIELD ID	SPACES
DEPT	00	EMPNUM	00
FNAME	00	LNAME	02
OVRHOURS	02	REGHOURS	02
TOTHOURS	02	WAGERATE	02
TOTPAY	05		

Field Sequence

Field sequence is assigned in the following screen. Field names are presented in alphabetical order, and all fields are listed as on data line one; since only one data line is desired for this report, this default value is retained for all fields. The sequence numbers for the fields within data line one, then, are modified according to their order in the report. The fields not to be printed in the report, EMPNUM and HOURS, are assigned sequence numbers of 99. This screen appears as follows.

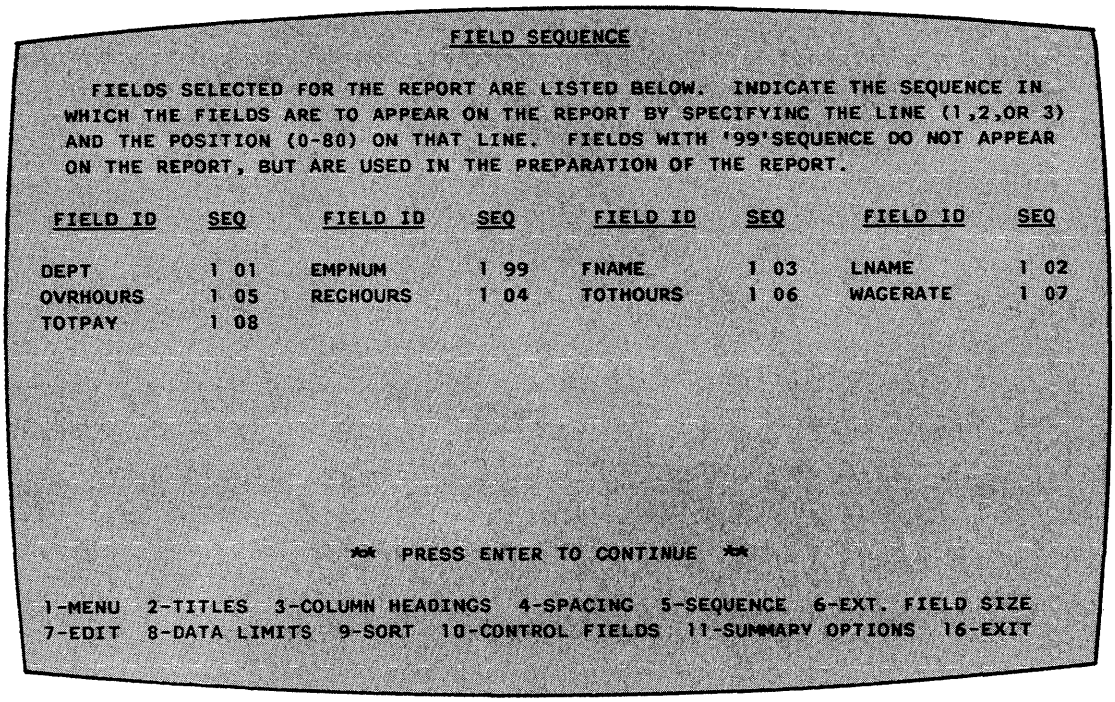


Figure 8-12. Field Sequence Screen

External Field Size

The External Field Size screen permits the user to modify the external sizes of the fields, if desired. As explained earlier, LNAME is truncated to 10 places from 20, FNAME is truncated from 10 to 9, and TOTPAY is truncated from 10 to 9 to eliminate the place created for its sign. The final external field sizes are as follows.

FIELD ID	SIZE	FIELD ID	SIZE
DEPT	2	LNAME	10
FNAME	9	REGHOURS	2
OVRHOURS	2	TOTHOURS	3
WAGERATE	5	TOTPAY	9

Data Edit Options

The Data Edit Options screens allow the user to modify the external representations of the numeric fields. The first Data Edit Options screen displays the existing representations (e.g., 99999.99- for TOTPAY), and requests the user to specify which field representations are to be altered. Since the representations of REGHOURS, OVRHOURS, TOTHOURS, and TOTPAY are to be altered, an "X" is typed next to each of these field names and ENTER is pressed.

The second Data Edit Options screen requests the user to specify the data edit options desired for the specified fields. Since zero suppression is desired for REGHOURS, OVRHOURS, and TOTHOURS, "Z1" is entered under the Suppress Zeroes column to indicate that only the rightmost position is not to be zero-suppressed for each of these fields. Since zero suppression is also desired for TOTPAY, "Z3" is similarly entered. Additionally, TOTPAY's sign is removed by spacing over the default value of "9-" in the Sign Control column, and its dollar sign is inserted by keying "$$" under the Dollar Sign column to float the dollar sign. Also, since commas are to be inserted into TOTPAY values, a "Y" is entered in the Commas column. This screen now appears as follows.

```
                          OUTPUT DATA EDIT OPTIONS

     THE NOMINAL DATA EDIT FORMATS ARE DISPLAYED FOR EACH FIELD THAT IS TO
          BE MODIFIED.  FOR AN EXPLANATION OF THE CODES, PRESS PF14.

                (*n,Zn)                          (.n)    n(.,/*- )  ($$,$9)  (Y OR
                SUPPRESS      (CR,DB,9-)        DECIMAL    SPECIAL    DOLLAR     N )
  FIELD ID      ZEROES        SIGN CONTROL      CARRY      INSERTION  SIGN     COMMAS
 REGHOURS       Z1            **                .0         ** ** **   **         N
 OVRHOURS       Z1            **                .0         ** ** **   **         N
 TOTHOURS       Z1            **                .0         ** ** **   **         N
 TOTPAY         Z3            **                .2         ** ** **   $$         Y

                              ** PRESS ENTER TO CONTINUE **

 1-MENU  2-TITLES  3-COLUMN HEADINGS  4-SPACING  5-SEQUENCE  6-EXT. FIELD SIZE
 7-EDIT  8-DATA LIMITS  9-SORT  10-CONTROL FIELDS  11-SUMMARY OPTIONS  16-EXIT
```

Figure 8-13. Data Edit Options Screen

Note that zero-suppression was not selected for the numeric fields DEPT and WAGERATE. This is because the department numbers are always referred to as two digit numbers and, therefore, the zero is desired; wage rates are always three digit numbers and, therefore, never contain leading zeros.

Data Limits

The Data Limits screen permits the user to specify limits for the values of fields to be used in this report. For this example, all of the data is to be printed; therefore, this screen is not altered.

File Sort

The File Sort screen requests the names of the fields on which the data is to be sorted. As discussed earlier, these fields are entered as follows.

FIELD ID	LEVEL	ASCENDING/DESCENDING
DEPT	1	A
LNAME	2	A
FNAME	3	A

Control Fields

The Control Fields screen requests the name(s) and level(s) of the control field(s). As discussed earlier, this report has only one control field, DEPT, which is level 1. Additionally, three lines are to be skipped after each control break, and the value for DEPT is to be printed only once for each control section. This screen appears as follows.

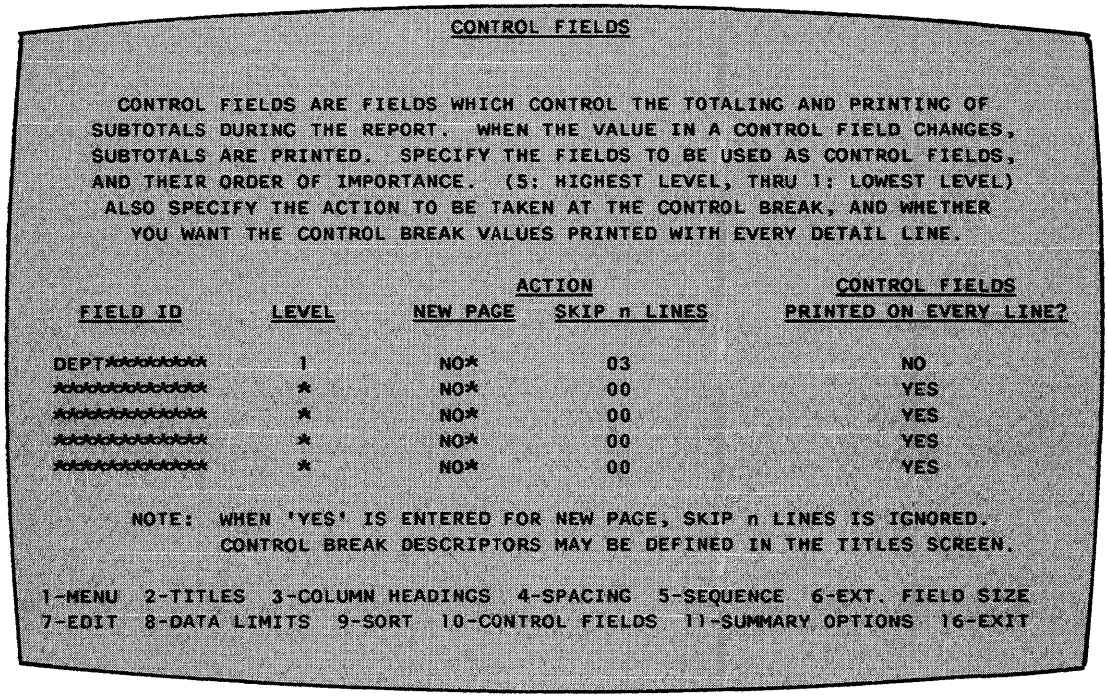


Figure 8-14. Control Fields Screen

Report Summary Options

The final screen, the Report Summary Options screen, requests the user to indicate which, if any, summary values are to be printed in this report. Since a total value is to be printed for TOTPAY, an "X" is keyed into the TOTAL column for the field TOTPAY. After ENTER is keyed, the user is returned to the Report Definition Options screen. Pressing PF 16 returns the user to the REPORT Main menu.

At this time, the report can be modified, displayed, or printed by pressing either PF3, Modify Report Attributes, or PF4, Print a Report.

8.4.4 Printing the Report

Since it is often best to examine a report before it is printed, PF4, Print a Report, is pressed from the REPORT Main menu. The subsequent screen requests the Report ID (Payroll), the report date (used if specified for the report), and the output device (DISPLAY for examination). Additionally, this screen requests whether or not the count option is to be used (NO); whether or not only summary lines are to be printed (since the data as well as the summary lines are to be examined, this response is NO); the number of lines per page (55); which data lines (and their order) are to be printed (123); and the print line spacing (000, since no blank lines are to be printed between data lines). After these parameters have been entered, the report is displayed on the screen.

If the displayed report is satisfactory, the user can print the report by returning to the Report Main menu (PF16), reselecting PF4, Print a Report, and respecifying the report. This time the user specifies that the output should go to the printer. If the displayed report is not satisfactory, it can be altered by selecting PF3, (Modify Report Attributes) from the Report Main menu, after which it can again be examined.

The completed report PAYROLL, run from the control file PAYROLL (in IDCTL) and the data file PAYROLL (in IDDATA), appears as follows.

EMPLOYEE WAGES

DEPT	EMPLOYEE		HOURS	OVERTIME	TOTAL HOURS	WAGE RATE	TOTAL PAY
08	ADAMS,	SUSAN B.	40	8	48	$4.75	$247.00
	JONES,	FRED A.	40	0	40	$5.00	$200.00
	ROBERTS,	ANNE R.	40	5	45	$5.25	$249.38
						TOTAL WAGES:	$696.38
12	FAULKNER,	CINDY	40	3	43	$5.50	$244.75
	KENNERLY,	SARAH F.	40	0	40	$3.50	$140.00
	SMITH,	THOMAS D.	35	0	35	$5.00	$175.00
						TOTAL WAGES:	$559.75
15	BUTLER,	JEAN K.	40	15	55	$6.00	$375.00
	HARRIGAN,	MAUREEN	40	0	40	$5.50	$220.00
						TOTAL WAGES:	$559.75
							$1,851.13

Figure 8-15. Printed Sample Report

APPENDIX A
FILE MANAGEMENT UTILITY GETPARMS

A.1 INTRODUCTION TO GETPARMS

The VS Operating System supports a supervisor call routine (SVC), the "GETPARM" SVC, which solicits and accepts run-time parameter information, and displays and awaits acknowledgment of run-time messages. GETPARM-generated prompts are displayed on the workstation screen during normal execution. These prompts solicit parameter information from the user or from a controlling procedure. Values entered from either source are verified for validity. If the values entered are not acceptable, the GETPARM SVC responds with an error message.

GETPARM processing is distinguished from other methods of obtaining run-time information primarily because it can interface with a procedure (see the VS Procedure Language Reference). A procedure is the preferred source of information for a GETPARM request; thus, GETPARM prompts never appear on the workstation screen when they are satisfied by a Procedure language ENTER statement. Therefore, the interaction of the user with a program at run-time can be precisely controlled by the procedure writer. GETPARM requests are used wherever possible by the VS system programs to solicit parameter information. This enables the user to run system programs with little or no user interaction by supplying most or all of the required program parameters from procedures.

A.2 THE STRUCTURE OF A GETPARM

Each GETPARM request in a program is identified with a parameter reference name (prname). The prname for each request is, in general, unique within that program. System programs generally observe certain conventions when identifying GETPARM requests with prnames. For example, a GETPARM request soliciting information for an input file is usually identified with the prname INPUT, while a GETPARM soliciting parameters for an output file is named OUTPUT.

Many GETPARM requests for information contain one or more modifiable fields into which the user (or a procedure) can enter information. Each of these fields is labelled with a name called a keyword. When a GETPARM request is displayed, the keyword for each field provides a description of the information to be supplied for that field. Again, certain conventions are commonly used in keyword naming; for example, a request for a file name often uses the keyword FILE. Also, many GETPARM requests solicit a PF key response (such as 16 = Exit Program). There is no keyword associated with a PF key choice; only the PF key number itself is specified.

A.3 ASSOCIATING A PROCEDURE WITH A GETPARM

Within a procedure, each ENTER or DISPLAY statement supplies parameters for a single GETPARM request. To associate a given ENTER or DISPLAY statement with a specific GETPARM request, the prname of the request must be specified in the statement. (GETPARM requests issued by a user program may, of course, be assigned any prname desired by the programmer. For user-defined GETPARM requests, modifiable fields and the keywords identifying them are specified by the GETPARM issuer.)

When parameters are supplied by a procedure, keywords must be used in the ENTER or DISPLAY statement to associate the specified values with the fields to which they are to be assigned in the GETPARM request. In this case, values associated with keywords in the procedure statement are passed to the corresponding keyword-identified fields in the GETPARM request. (Keywords must, of course, be correctly spelled in the procedure statement.) Fields that are not assigned new values retain their default values.

An example of a procedure that runs the DATENTRY utility is provided below. By comparing this procedure to the list of prnames and keywords for DATENTRY, it can be seen that certain default keyword values have been used, since those keywords are not listed (e.g., CTLLIB and CTLVOL for prname INPUT). Also, PF key options have been selected for certain other prnames (e.g., for prname OPTIONS: PF2, Create a Data File, was selected).

```
PROC RUN DATENTRY
RUN DATENTRY
ENTER INPUT FILE = TEST, LIBRARY = IDCTL,
      VOLUME = SYSTEM, CTLFILE = TEST
ENTER OPTIONS  2
ENTER OPTIONS 16
RETURN
```

Refer to the VS Procedure Language Reference for more information on the Procedure language and the use of GETPARM requests.

A.4 FILE MANAGEMENT UTILITY GETPARM REQUESTS

The following pages list the prnames, keywords, options, and default values used by GETPARM requests for the VS File Management Utilities. The GETPARMS are listed by utility, and the utilities are organized alphabetically. Please note that when PF keys are used as options for a prname, no keyword is listed. Similarly, when the ENTER key is used as a PF key option, "0" is listed; otherwise use of the ENTER key is assumed.

CONDENSE

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULTS
PARMFILE	PARMFILE	8		
	PARMLIB	8		User ID & COND
	PARMVOL	6		
	PF Keys*		0 = Continue	
			16 = Exit CONDENSE	
FUNCTION	PF Keys*		2 = Create parameter file	
			3 = Modify existing parameter file	
			4 = Create condensed file	
			5 = Run REPORT	
			16 = Exit to respecify parameter file	
USEREXIT	FILE	8		
	LIBRARY	8		
	VOLUME	6		
CONTROL	CTLFILE	8		
	CTLLIB	8		User ID & CTL
	CTLVOL	6		
INPUT	INFILE	8		
	INLIB	8		
	INVOL	6		
OUTPUT	OUTFILE	8		
	OUTLIB	8		
	OUTVOL	6		

* The keyword is not required.

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULTS
CTLFIL	FILE	8		
	LIBRARY	8		User ID+CTL
	VOLUME	6		INVOL
	PF Keys*		0 = CONTINUE	
			16 = EXIT	
OPTIONS	PF Keys*		2 = CREATE CTL FILE	
			3 = ADD RECORDS	
			4 = MODIFY CTL FILE	
			5 = DELETE RECORDS	
			6 = LIST RECORDS	
			7 = MAINTAIN TABLES	
			8 = COBOL COPYLIB	
			9 = DATENTRY	
			10 = REPORT	
			11 = EZFORMAT	
			12 = MODIFY FIELD UPDATE SEQUENCE	
			16 = EXIT	
DELETE	FIELD	8		
	PF Keys*		0 = CONTINUE	
			16 = EXIT	
HEADER	RECLEN	4	1-2040 (fixed)	
			1-2020 (variable)	
			1-2024 (compressed length)	
	KEYFIELD	8		
	ALTKEYS	2	0-16	0
	USEREXIT	8	0, USER1-USER10	0
	REPORT	1	0, 1	0
	UPDATE	1	0, 1	0
	DELETE	1	0, 1	0
	FILETYPE	1	F, V, C	F
	DSCRPTN1	60		
	DSCRPTN2	60		
	DSCRPTN3	60		
	PF Keys*		0 = CONTINUE	
			16 = EXIT TO CTL MENU	

* The keyword is not required.

CONTROL (Continued)

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULTS
CPYLIB	FILE	8		
	LIBRARY	8		
	VOLUME	6		
	PF Keys*		0 = CONTINUE 16 = EXIT	
EDFILE	FILE	8		Data file name
	LIBRARY	8		Data file library
	VOLUME	6		Data file volume
	RECORDS	7		
	RETAIN	3		
	RELEASE	3	YES or NO	YES
	FILECLAS	1	A-Z, #,$,@," "	#
	DEVICE	11	DISK	DISK
CTLFILE**	(OUTPUT default GETPARM)			
PRINTFL**	(PRINT default GETPARM)			

* The keyword is not required.

** See Section A.5.

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULTS
INPUT	FILE	8		
	LIBRARY	8		
	VOLUME	6		
	CTLFILE	8		
	CTLLIB	8		USER I.D. + CTL
	CTLVOL	6		Input volume
	PF Keys*		∅ = CONTINUE	
			16 = EOJ	
OPTIONS	PF Keys*		2 = CREATE DATA FILE	
			3 = ADD RECORDS	
			4 = MODIFY RECORDS	
			5 = DELETE RECORDS	
			6 = LIST RECORDS	
			16 = EXIT	
INDFILE	FILE	8		data file name
	LIBRARY	8		data file library
	VOLUME	6		data file volume
	RECORDS	7		500
	RETAIN	3		
	RELEASE	3	YES or NO	NO
	FILECLAS	1	A-Z, ∅, #, $	∅
	DEVICE	11	DISK	DISK
PATH	PATH	8		
	PF Keys**		∅ = CONTINUE	
			16 = EXIT	

* The keyword is not required.

DATENTRY (Continued)

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULTS
PATHS	PATH1	8		1st alternate key
	PATH2	8		2nd alternate key
	.	.		
	.	.		
	.	.		
	PATH 16	8		16th alternate key
	DUPS1	3	YES OR NO	YES
	.	.	.	.
	.	.	.	.
	.	.	.	.
	DUPS16	3	YES or NO	YES
	PF Keys*		0 = MAKE CHANGES 1 = RETURN TO HEADER SCREEN 16 = EXIT	

* The keyword is not required.

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULTS
OPTIONS	LANGUAGE	10	ASSEMBLER,A,COBOL,C, BASIC,B,RPG,R, D,MENU,M	
	PROCEDUR	3	YES,NO	YES
	PF Keys*		2 = New screen format 3 = Existing screen format	
INSCREEN	FILE	8		
	LIBRARY	8		User ID + SAVE
	VOLUME	6		
	DEVICE	11	DISK,NONE	DISK
FUNCTION	PF Keys*		2 = Save generated output only 3 = Save screen contents only 4 = Save screen contents and generated output 16 = Exit without saving any files	
SCREEN	FILE	8		
	LIBRARY	8		User ID + SAVE
	VOLUME	6		
CONTROL	FILE	8		
	LIBRARY	8		User ID + CTL
	VOLUME	6		
	PF Keys*		0 = Continue 1 = Switch to COBOL OPTION 2 = Invoke CONTROL 16 = Exit	
COPYLIBR	FILE	8		
	LIBRARY	8		User ID + COPY
	VOLUME	6		
PROGRAM	FILE	8		
	LIBRARY	8		
	VOLUME	6		

* The keyword is not required.

INQUIRY

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULTS
INPUT	FILE	8		
	LIBRARY	8		
	VOLUME	6		
	MODE	5	INPUT, SHARED	INPUT
	CHANGE	3	YES, NO	NO
	OPTION	7	DISPLAY, EXTRACT	DISPLAY
	PF Keys*		⅋ = CONTINUE 16 = Exit	
CONTROL	FILE	8		
	LIBRARY	8		User ID & CTL
	VOLUME	6		
	PF Keys*		⅋ = CONTINUE 1 = Return to input data selection	
QUERY	DISPLAY		YES, NO	YES
	TITLE	40		
	LINE 1A	40		
	LINE 1B	39		
	LINE 2A	40		
	LINE 2B	39		
	LINE 3A	40		
	LINE 3B	39		
	LINE 4A	40		
	LINE 4B	39		
	LINE 5A	40		
	LINE 5B	39		
	LINE 6A	40		
	LINE 6B	39		
	LINE 7A	40		
	LINE 7B	39		
OUTPUT	FILE	8		
	LIBRARY	8		
	VOLUME	6		
	RECORDS	7		No. of records in interrogated file
	CONSEC	3	YES, NO	NO

* The keyword is not required.

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULTS
EOJ	PF Keys*		1 = New Query 2 = Save Query 3 = Create RDF 16 = Exit	
PROGRAM	FILE	8		
(PF 2)	LIBRARY	8		
	VOLUME	6		
	DISPLAY	3	YES, NO	YES
	ENDRUN	3	YES, NO	NO
	PF Keys*		0 = Continue 1 = Return	
RPTDEF	FILE	8		
(PF 3)	LIBRARY	8		User ID & RPT
	VOLUME	6		
	PF Keys*		0 = Continue 1 = Return	

* The keyword is not required.

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULT
FUNCTION	PFKEYS		2 = DEFINE REPORT 3 = MODIFY DEFINITION 4 = PRINT REPORT 16 = EOJ	
CONTROL	FILE	8		primary Control File name
	LIBRARY	8		USER I.D. + CTL
	VOLUME	6		primary Control File volume
CONTROL2	FILE	8		secondary Control File name
	LIBRARY	8		USER I.D. + CTL
	VOLUME	6		secondary Control File volume
RPTDEF	FILE	8		Report Definition File name
	LIBRARY	8		Report Definition File library
	VOLUME	6		Report Definition File volume
INPUT1	FILE	8		Primary Data File name
	LIBRARY	8		Primary Data File library
	VOLUME	6		Primary Data File volume
	MODE	6	INPUT or SHARED	
INPUT2	FILE	8		Secondary Data File name
	LIBRARY	8		Secondary Data File library
	VOLUME	6		Secondary Data File volume
	MODE	6	INPUT or SHARED	
OPTIONS	ID	8		
	DATE 8		current date	
	DEVICE	7	PRINTER/DISPLAY	PRINTER
	FILES	3	YES or NO	
	OPTION	3	YES/NO or	
	001 - 999	NO		
	ONLY	3	YES/NO	NO
	PAGE	2	05 - 99	55
	LINES	3	Combination of 1, 2, and 3	
	SPACING	3	Combination of 0-5.	
	PF Keys*		0 = CONTINUE 14 = EXPLANATIONS 16 = RETURN TO RPT MENU	
PRINT**	(PRINT default GETPARM)			

* The keyword is not required.

** Refer to Section A.5.

A.5 DEFAULT GETPARMS

Default GETPARMS are parameter requests that are not normally displayed to the user because they are already supplied default values. These default values can either be supplied by the system, as in the case of a user's print library name, or by values specified in the Set Usage Constants function of the Command Processor. To change the default values, specify the appropriate prname and keywords in a procedure.

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULT
OUTPUT	FILE	8		System Generated
	LIBRARY	8		# logon I.D. + WORK
	VOLUME	6		System Disk
	RECORDS	7		Number of Records
	RETAIN	3		0
	RELEASE	3	YES or NO	YES
	FILECLAS	1		#
	DEVICE	11	DISK, TAPE	DISK
PRINT	FILE	8		System Generated
	LIBRARY	8		# logon I.D. + PRT
	VOLUME	6		System Disk
	RECORDS	7		Number of Records
	RETAIN	3		0
	RELEASE	3	YES or NO	YES
	FILECLAS	1		#
	DEVICE	11	DISK, PRINTER, TAPE	DISK
	PRTCLASS	1		A
	FORM#	3		000

APPENDIX B
CONTROL FILE RECORD FORMATS

B.1 INTRODUCTION

The Control file is an indexed file with the key in bytes 3-10 and a record length of 130 bytes. It has fixed-length records and is stored in the library USERID + CTL (unless the user overrides this default).

The Control file contains File Descriptor Records, Alternate Key Records, Comment Records, and Field Descriptor Records. A File Descriptor Record consists of the File Name, File Type, Key Field, Report Code, Update Code, Delete Code, Record Length, User Exit (optional), and the Number of Alternate Keys. If an alternate key is specified, the Alternate Key Record contains the Alternate Key Name and whether or not this key allows duplicates. An optional Comment Record consists of the File Description. A Field Descriptor Record consists of the Field Name, Internal Format, Internal Length, Starting Position, Occurrences, Zero Suppress Code, Decimal Insert Code, Sign Control Code, Dollar/Comma Code, External Length, Report Code, Update Code, Decimal Positions Code, Binary Edit Code, Update Sequence, Validation Type, Table Name, Low Range, High Range, Packed Digits, Cumulative Field Name, Field Alias, Display Code, and Report Field Length.

B.2 HEADER RECORD

The format of the Header Record is as follows.

Byte Number	Field Name	Permissible Values
1-2	Not Used	ASCII zeroes
3-10	Header Label	Must be "␣HEADER␣"
11	File Type	F - Fixed-Length Records V - Variable-Length Records C - Compressed Records
12-19	Key Field	Name of key field for indexed file. If blank, consecutive file assumed.
20	Report Code	0 = Allowed 1 = Not Allowed

Byte Number	Field Name	Permissible Values
21	Update Code	0 = Allowed 1 = Not Allowed
22	Deletion Code	0 = Allowed 1 = Not Allowed
23-26	Record Length	Record length of data file (1-2040). If variable-length records, max. rec. length.
27-29	Not Used	
30-37	User Exit	Name of User Exit Subroutine, USER1 - USER10. Total number of alternate keys in data file.
38-39	Number of Alternate Keys	Numeric values, 0-16
40-130	Not Used	

B.3 ALTERNATE KEY RECORD

The format of the Alternate Key Record is as follows.

Byte Number	Field Name	Permissible Values
1-2	Not Used	
3-10	Record Name	Must be "bKEY1bbb" (KEY 2 for second record, etc. to KEY 8.)
11-18	Alternate Key Name 1	Valid field name
19	Allow Duplicates 1	Y(YES) N(NO)

The above fields are repeated 7 times, with their byte positions as follows.

20-27	Alternate Key Name 2
28	Allow Duplicates 2
29-36	Alternate Key Name 3
37	Allow Duplicates 3
38-45	Alternate Key Name 4
46	Allow Duplicates 4

Byte Number	Field Name	Permissible Values
47-54	Alternate Key Name 5	
55	Allow Duplicates 5	
56-63	Alternate Key Name 6	
64	Allow Duplicates 6	
65-72	Alternate Key Name 7	
73	Allow Duplicates 8	
74-81	Alternate Key Name 8	
82	Allow Duplicates 8	
83-130	Not Used	

NOTE

There may be up to two of these records.

B.4 COMMENT RECORD

The format of the Comment Record(s) (File Description) is as follows.

Byte Number	Field Name	Permissible Values
1-2	Not Used	
3-10	Comment Descriptor	Must be "!1␣COMM␣", "!2␣COMM␣", or "!3␣COMM␣".
11-70	Comment	Alphanumeric Data
71-130	Not Used	

NOTE

There may be up to three of these records.

B.5 FIELD DESCRIPTOR RECORD

The format of the Field Descriptor Record is as follows.

Byte Number	Field Name	Permissible Values
1-2	Not Used	
3-10	Field Name	1-8 character field name. Must begin with alphabetic character and not contain any embedded blanks.

Byte Number	Field Name	Permissible Values
11	Internal Format	B - Binary C - Character P - Packed Decimal Z - Zoned Decimal
12-14	Internal Length	Number of bytes required for internal representation (1-67).
15-18	Starting Position	Starting position of field. Must be less than record length.
19-20	Occurrences	Number of occurrences of this field (01-99).
21	Zero Suppress Code (Used by REPORT)	0 - No Zero Suppress 1 - Suppress Leading Zeroes 2 - Protect
22	Decimal Insert Code (Used by REPORT)	0-9 - Number of decimal positions to print out for report utility.
23	Sign Control Code (Used by REPORT)	0 - No Sign Control 1 - Trailing Minus Sign 2 - CR on Minus (Trailing) 3 - DB on Minus (Trailing)
24	Dollar/Comma Code (Used by REPORT)	0 - No Dollar or Comma Edit 1 - Comma Edit for Numeric Field 2 - Dollar Edit for Numeric Field 3 - Dollar and Comma Edit for Numeric Field
25-27	External Length	1-67, calculated by CONTROL program from Internal Byte length, or entered by the user. If character field, external field length = number of disk positions. If zoned decimal field (format = Z), external field length = number of disk positions + 1 (for sign) + 1 (for decimal point if number of decimal positions not equal to zero). If packed decimal field (format = P), external field length = (number of disk positions * 2) + 1 (for decimal point if number of decimal positions not = zero).

Byte Number	Field Name	Permissible Values
	External Length, Cont.	If binary field, (format = B), and binary edit code is equal to 0 (see below) for hexadecimal field; external field length = number of disk positions *2. If binary field, (format = B), and binary edit code is equal to 1 (see below), for conversion to binary field. If number of disk positions = 1, external field length = 2. If number of disk positions = 2, external field length = 3. If number of disk positions = 3, external field length = 5. If number of disk positions = 4, external field length = 8.
28	Report Code	0 - This field can be reported on using REPORT 1 - This field cannot be reported on using REPORT
29	Update Code	0 - This field can be updated using DATENTRY 1 - This field cannot be updated using DATENTRY 2 - Display this field without allowing update in DATENTRY
30	Decimal Positions Code	Number of decimal positions; numeric 0-9
31	Binary Edit Code	(Used only if field format = B.) 0 - This binary field is represented in hexadecimal 1 - This binary field is represented as decimal
32	Not Used	
33-34	Update Sequence (Used by DATENTRY)	Sequence in which this field will appear for formatted screen in DATENTRY.

Byte Number	Field Name	Permissible Values
35-36	Validation Type	Validation criteria for field for DATENTRY. T - An external table is used. R - This field is checked for being within a range. CF - This field is accumulated in a predefined field. Both this field and the predefined field must be numeric (format = P or Z). DF - This field is a date field. DATENTRY will display the system date (MMDDYY) for this field when records are added to the data file.
37-42	Table Name	1-6 character name of external validation table for this field. Used only if validation type = Table.
43-58	Low Range	Low range for this field. Used for range checking by DATENTRY. Used only if validation type = R.
59-74	High Range	High range for this field. Used for range checking by DATENTRY. Used only if validation type = R.
75-76	Packed Digits	Number of packed digits (if internal format = P). This is the (number of disk positions *2) - 1.
77-84	Cumulative Field Name	Name of cumulative source field. Used if validation type = CF. The cumulative field name must be an existing numeric field defined on the Control file.
85-115	Field Alias	An alternate name for the fields to be used by the INQUIRY program.
116-124	Not Used	
125	Display Code	0 - Pseudoblanks appear 1 - Data remains on screen 2 - New data added, but no modification

Byte Number	Field Name	Permissible Values
126-128	Report Field Length	Total number of Field Positions required for Report Format (includes External Length, Signs, Symbols, etc.).
129	Default FAC Character	Any valid Field Attribute Character (refer to the VS Principles of Operation).
130	Not Used	

APPENDIX C
REPORT FILE RECORD FORMATS

C.1 INTRODUCTION

The Report Definition file is an indexed file with the key in bytes 1-13 and a record length of 87 bytes. It has compressed records and is stored in the library USERID + RPT (unless the user overrides this default).

The Report Definition File consists of a Header Record, a Data Files Record (containing information concerning the input data file(s)), a Control Files Record (containing information about the control file(s)), Sort Records (containing the fields by which the data is sorted), a Control Fields Record (containing control break information), Title Descriptor Records, Data Limits Records (containing data selection criteria), New Field Records (containing information on any new fields that are specified), Field Descriptor Records (containing the printing specifications for individual fields), and a User Exit record (containing the subroutine name and its file, library, and volume names).

Many of these records, or parts of them, are unnecessary for specific reports and, therefore, are not created. This is true of the Control Files Record (file location may be obtained from the Data Files Record, library may be obtained from IDCTL, and volume may be obtained from SET USAGE CONSTANTS). Also unnecessary for specific reports are the Sort Records (if no sorting is performed), the Control Fields Record (if no control breaks are selected), some of the Title Descriptor Records (only the first report title is required), and the New Field Records (if no New Fields are specified). Note also that if only part of a record is required, the remainder is blank.

C.2 HEADER RECORD

The format of the Header Record is as follows.

Byte Number	Field Name	Permissible Values
1-13	Key to Header Record	Must be "␣␣HEADER␣␣␣␣␣".
14	Secondary File Switch	1 - Secondary file is used 0 - No secondary file is used

Byte Number	Field Name	Permissible Values
15	Sort Switch	1 - Sorting performed 0 - No sorting performed
16	Data Limits Switch	1 - Data Limits specified 0 - No Data Limits specified
17	Control Fields Switch	1 - Control Fields used 0 - No Control Fields used
18	Column Subheadings Switch	1 - Column subheadings specified 0 - No column subheadings specified
19	New Fields Switch	1 - New Fields specified 0 - No New Fields specified
20-21	Print Line Length	Packed format, numeric value. Legal range 1-132.
22-23	Number of Fields	Packed format, numeric value. Legal range 1-40. Total number of fields chosen for REPORT (includes non-printed fields).
24-25	Number of Print Fields	Packed format, numeric value. Legal range 1-40. Total number of fields selected for printing.
26-27	Second Print Line Length	Packed format, numeric value. Range 1-132.
28-29	Third Print Line Length	Packed format, numeric value. Range 1-132.
30	User Exit Switch	1 - User Exit used 0 - No User Exit used
31-87	Not Used	

C.3 DATA FILES RECORD

The format of the Data Files Record is as follows.

Byte Number	Field Name	Permissible Values
1-13	Key to Data Files Record	Must be "␣␣FILES␣␣␣␣".
14-21	Primary Data File Name	1-8 character file name. Must begin with an alphabetic character and cannot contain any embedded blanks.

Byte Number	Field Name	Permissible Values
22-29	Primary Data File Library	1-8 character library name. Must begin with an alphabetic character and cannot contain any embedded blanks.
30-35	Primary Data File Volume	1-6 character volume name. Must be an existing volume name.
36-43	Secondary Data File Name	1-8 character file name. Must begin with an alphabetic character and cannot contain any embedded blanks.
44-51	Secondary Data File Library	1-8 character library name. Must begin with an alphabetic character and cannot contain any embedded blanks.
52-57	Secondary Data File Volume	1-6 character volume name. Must be an existing volume name.
58-65	Key to Secondary Data File	A field from the primary file to be used as the key to the secondary file. Any valid field name found in primary file.
66-67	Occurrences	Numeric value, legal range 1-99 or blank. If the occurence of the field chosen from the primary file is greater than 1, then this value indicates the array subscript of the field to be used as the key from the primary file. Otherwise, this value is blank.
68-69	Not Used	Always blank.
70-87	Not Used	

C.4 CONTROL FILES RECORD

The format of the Control Files Record is as follows.

Byte Number	Field Name	Permissible Values
1-13	Key to Control Files Record	Must be "bbFILESCTLbbb".

Byte Number	Field Name	Permissible Values
14-21	Primary Control File Name	1-8 character file name. Must begin with an alphabetic character and cannot contain any embedded blanks.
22-29	Primary Control File Library	1-8 character library name. Must begin with an alphabetic character and cannot contain any embedded blanks.
30-35	Primary Control File Volume	1-6 character volume name. Must be an existing volume name.
36-43	Secondary Control File Name	1-8 character file name. Must begin with an alphabetic character and cannot contain any embedded blanks.
44-51	Secondary Control File Library	1-8 character library name. Must begin with an alphabetic character and cannot contain any embedded blanks.
52-57	Secondary Control File Volume	1-6 character volume name. Must be an existing volume name.
58-87	Not Used	

C.5 SORT RECORDS

The format of Sort Record 1 is as follows.

Byte Number	Field Name	Permissible Values
1-13	Key to Sort Record 1	Must be "␣␣SORT1␣␣␣␣␣".
14-21	Field Name 1	1-8 character field name. Must begin with an alphabetic character and cannot contain any embedded blanks.
22-23	Occurrences 1	Numeric Value, legal range 1-99 or blank.
24	Order Code 1	A - Ascending sort order D - Descending sort order

These last three fields are repeated 5 more times as needed with their byte positions as shown below.

25-32	Field Name 2
33-34	Occurrences 2
35	Order Code 2
36-43	Field Name 3
44-45	Occurrences 3
46	Order Code 3
47-54	Field Name 4
55-56	Occurrences 4
57	Order Code 4
58-65	Field Name 5
66-67	Occurrences 5
68	Order Code 5
69-76	Field Name 6
77-78	Occurrences 6
79	Order Code 6
80-87	Not Used

The format of Sort Record 2 is as follows.

Byte Number	Field Name	Permissible Values
1-13	Key to Sort Record 2	Must be "bbSORT2bbbbbb".
14-23	Field Name 7	1-8 character field name. Must begin with an alphabetic character and cannot contain any embedded blanks.
24-25	Occurrences 7	Numeric value, legal range 1-99 or blank.
26	Order Code 7	A - Ascending sort order D - Descending sort order

These fields are an extension of the record Sort Record 1. These last three fields are repeated once, with byte positions shown below.

27-34	Field Name 8
35-36	Occurrences 8
37	Order Code 8
38-87	Not Used

C.6 CONTROL FIELDS RECORD

The format of the Control Fields Record is as follows.

Byte Number	Field Name	Permissible Values
1-13	Key to Control Fields Record	Must be "bCNTRLbFLDSbb".
14-21	Field Name 1	1-8 character field name. Must begin with an alphabetic character and cannot contain any embedded blanks.
22-23	Occurrences 1	Numeric value, legal range 1-99 or blank.
24-25	Control Break Action Code 1	YE - Page eject follows Control Break. 0-99 - Numeric values: specifies number of lines to be skipped following Control Break.
26	Field Origin Code 1	1 - Primary File 2 - Secondary File 3 - New Field
27	Control Field Repeat Code 1	1 - Control Field name repeated on every data line. 0 - Control Field name printed only on first data line of Control Break section.

These last five fields are repeated 4 more times with byte positions shown below.

28-35	Field Name 2
36-37	Occurrences 2
38-39	Control Break Action Code 2
40	Field Origin Code 2
41	Control Field Repeat Code 2
42-49	Field Name 3
50-51	Occurrences 3
52-53	Control Break Action Code 3
54	Field Origin Code 3
55	Control Field Repeat Code 3

56-63	Field Name 4
64-65	Occurrences 4
66-67	Control Break Action Code 4
68	Field Origin Code 4
69	Control Field Repeat Code 4
70-77	Field Name 5
78-79	Occurrences 5
80-81	Control Break Action Code 5
82	Field Origin Code 5
83	Control Field Repeat Code 5
84-87	Not Used

C.7 TITLE DESCRIPTOR RECORDS

The format of the Title Discriptor Records is as follows.

Byte Number	Field Name	Permissible Values
1-13	Key to Title Records	The three types of titles have different key values: Report Titles: "bH1-TITLEbbbb" "bH2-TITLEbbbb" "bH3-TITLEbbbb" Page Titles: "bH4-PAGEHbbbb" "bH5-PAGEHbbbb" "bH6-PAGEHbbbb" Control Break Descriptors: "bHC1CNTRLbbbb" "bHC2CNTRLbbbb" "bHC3CNTRLbbbb" "bHC4CNTRLbbbb" "bHC5CNTRLbbbb"
14-73	Title	For Report Titles and Page Titles: 60 character alpha-numeric strings. For Control Break Descriptors: 40 character alphanumeric strings, followed by a 3-character numeric value for the Control Break Descriptor column position, followed by an unused space. Bytes — Use 14-53 — 40 char. string 54-56 — 3 char. string 57-87 — Not Used

Byte Number	Field Name	Permissible Values
74-87	Not Used	

NOTE

There may be up to 11 title records with all of the possibilities listed above (only the first is required). The key names given above correspond to the three types of titles noted above and have different title values in byte positions 14-73 as shown.

C.8 DATA LIMITS RECORDS

For a given report, it is possible to have up to 10 sets of Data Limits (designated by the values C - L in the Record Type Indicator field, byte position 1). Up to four records may exist for each Record Type Indicator (i.e., each field), depending on the number of fields/constants included in the data limits specifications. The first three records contain 3 fields and/or constants, and the last record contains only 1. The format of the Data Limits Records is as follows.

Byte Number	Field Name	Permissible Values
1	Record Type Indicator	Must be in the range C - L. The data limits sets begin with the letter C, and continue in alphabetic order up to L.
2-9	Field Name	1-8 character field name. Must begin with an alphabetic character and cannot contain embedded blanks.
10-11	Occurrences	Numeric value, legal range 1-99, or blank.
12	Field Origin Code	Indicates origin of field for which data limits are specified. 1 - Primary file 2 - Secondary file 3 - New field
13	Record Number	Numeric value, legal range 0-3. Identifies each of the four records for a given Record Type Indicator value (i.e., data field).

Byte Number	Field Name	Permissible Values
14-15	Operand Code 1	Operational connector between Field/Constants. Six options: GT - greater than LT - less than EQ - equal to GE - greater than or equal to LE - less than or equal to NE - not equal to

The next 20 bytes may contain two different types of information, depending upon the value in the field "Literal Indicator" (in byte position 35). If the Literal Indicator value is X, then byte positions 16 - 34 contain the information for a field that is to be used in the Data Limits Record. If the Literal Indicator value is not X, then the information in bytes 16-35 is assumed to be literal input to the Data Limits Record. These two types appear as follows.

Byte Number	Field Name	Permissible Values
	Field Information	
16-23	Field Name 1	Name of first field to be used in the Data Limits Record. Valid existing file name.
24-27	Occurrences 1	Numeric value, legal range 01-99, in parentheses (e.g., "(01)").
28	Field Origin Code 1	1 - Primary File 2 - Secondary File 3 - New Field
29-34	Not Used	
35	Literal Indicator 1	Must be "X" to indicate that the above is field information.
	Literal Information	
14-33	Literal Input	Byte position 35 must not be "X" to indicate literal information. Character: Input delimited by double quotation marks. Numeric: Maximum 15 digits (includes sign, if used, plus decimal point).

The rest of the record is as follows.

36	Logical Connector 1	Logically connects operands. Two options: A - And O - Or

These last 7 fields (bytes 16-36) are repeated twice with their byte positions as shown.

Byte Number	Field Name	Permissible Values
37-38	Operand Code 2	
39-46	Field Name 2	
47-50	Occurrences 2	
51	Field Origin Code 2	
52-57	Not Used	
58	Literal Indicator 2	
59	Logical Connector 2	
60-61	Operand Code 3	
62-69	Field Name 3	
70-73	Occurrences 3	
74	Field Origin Code 3	
75-80	Not Used	
81	Literal Indicator 3	
82	Logical Connector 3	

The end of the record is as follows.

83	Data Limits Set Connector	Logically connects set of data limits with its subsequent set. This field occurs only for the first of the four possible records for a given "Record Type Indicator" value (e.g., when the "Record Number" Value is "0"). There are two options: A - And O - Or
84-87	Not Used	

C.9 NEW FIELD RECORDS

The format of the New Field Records is as follows.

Byte Number	Field Name	Permissible Values
1	Record Type Indicator	Must be "R".
2-9	New Field Name	1-8 character field name. Must begin with an alphabetic character and cannot contain embedded blanks.
10-11	Record Number	Up to four records may exist in the definition of a New Field, the first three of which contain 3 fields and/or literals, and the last of which contains only 1. These records are identified by their Record Number as follows. 00 - First record 01 - Second record 02 - Third record 03 - Fourth record
12-13	New Field Number	Identifies the sequence in which the New Fields were created. Packed format numeric value, legal range 0 - 9.

The following 20 bytes may contain two different types of information, depending on the value in the field "Literal Indicator" (in byte position 33). If the Literal Indicator value is X, then the byte positions 14-32 contain the information for a field that is to be used in calculating the New Field. If the Literal Indicator value is not X, then the information in bytes 14-33 is assumed to be literal input to the New Field. These two modes appear as follows.

Field Information

14-21	Field Name 1	Name of first field to be used in creating New Field. Valid existing field name (may be a previously defined New Field).
22-25	Occurrences 1	Numeric value, legal range 1-99, in parentheses (e.g., "(01)"), or blank.
26	Field Origin Code 1	1 - Primary File 2 - Secondary File 3 - New Field
27 - 32	Not Used	

Byte Number	Field Name	Permissible Values
33	Literal Indicator 1	Must be "X" to indicate above field information.
	Literal Information	
14-33	Literal Input	Byte position 33 must not be "X" to indicate literal information. There are two types of literal input: Character: Input delimited by double quotation marks. Numeric: Maximum 15 digits (includes sign, if used), plus decimal point.

The rest of the record is as follows.

Byte Number	Field Name	Permissible Values
34	Operation 1	Indicates operation to be performed between already specified value (in bytes 14-33) and the next value to be specified. Options: & -Concatenates character information. +,-,/,* -For numeric information. blank -Indicates end of creation sequence.

These last 6 fields (bytes 14-34) are repeated either once or twice, as needed, with their byte positions as shown below.

35-42	Field Name 2
43-46	Occurrences 2
47	Field Origin Code 2
48-53	Not Used
54	Literal Indicator 2
55	Operation 2
56-63	Field Name 3
64-67	Occurrences 3
68	Field Origin Code 3
69-74	Not Used
75	Literal Indicator 3
76	Operation 3

The end of the record (i.e., information in bytes 77-81) is used only for the first of the four possible records (e.g., when the Record Number Value is 00).

Byte Number	Field Name	Permissible Values
77	New Field Type	C = Character P = Numeric
78-79	Internal New	Packed numeric value. Computed Field Length internally based upon specified New Field Length. Two types: Numeric: Always 8 bytes (for maximum calculation precision). Character: 1-132 byte range (same as specified length).
80	Decimal Positions	If New Field Type = P, numeric value, legal range 0 - 9. If New Field Type = C, blank.
81-87	Not Used	

C.10 FIELD DESCRIPTOR RECORD

The format of the Field Descriptor Record is as follows.

Byte Number	Field Name	Permissible Values
1	Record Type Indicator	Indicates on which print line the field is to appear. Z indicates line 1, Y indicates line 2, and X indicates line 3.
2-3	Field Sequence	Numeric values 1-40, 98 and 99.
4-11	Field Name	1-8 character file name. Must begin with an alphabetic character and cannot contain any embedded blanks.
12-13	Occurrences	Any subscript number. Numeric values 1-99, or blank if not part of an array.
14-38	Column Heading	Alphanumeric.
39-63	Column Subheading	Alphanumeric.
64-65	Spaces Before	Packed format; numeric values Fields 0-99.

Byte Number	Field Name	Permissible Values
66-67	External Size	Packed format; numeric values. Defaults to same value as CONTROL External Field Length.

NOTE

Bytes 68-84 are used for numeric fields only.

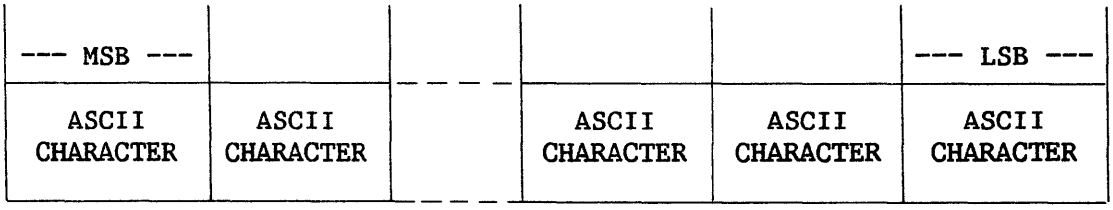
68-69	Zero Suppress Code	Blank - no zero suppression. *n - asterisk protection where n = 0-9. Zn - zero suppression where n = 0-9.
70-71	Sign Control Code	blank - No signs used 9- - Trailing minus sign CR - CR on minus (trailing) DB - DB on minus (trailing)
72	Decimal Carry	Numeric values 0-9 or blank.
73-78	Special Insertions 1, 2, and 3	Allows insertion of three special characters: nx, where n = 1-9 and x = ".", ",", "/", "*", "_", and blank.
79	Dollar Sign Code	0 - No dollar signs 1 - Floating dollar signs 2 - Fixed dollar signs
80	Comma Code	0 - No commas 1 - Commas inserted
81	Total Code	X - Totals printed at control breaks at end of report b - Totals not printed
82	Maximum Code	X - Maximum values printed blank b - maximums not printed
83	Mimimum Code	X - Minimum values printed blank b - minimums not printed
84	Average Code	X - Average values printed blank b - averages not printed
85	File Type or New Field Code	1 - Primary file 2 - Secondary file 3 - New field
86-87	Not Used	ASCII Zeros

APPENDIX D
DATA FORMATS

There are three types of data representations commonly used by the VS File Management Utilities. These data formats are described in greater detail in the VS Principles of Operation.

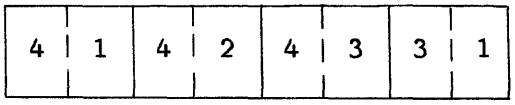
Character

In the character format, each character of an item is represented by a single byte in ASCII code. The maximum length of a character format item is 67 characters updatable, or 132 characters nonupdatable. The most significant byte and least significant byte are labelled as MSB and LSB, respectively. Character format can be diagrammed as follows.



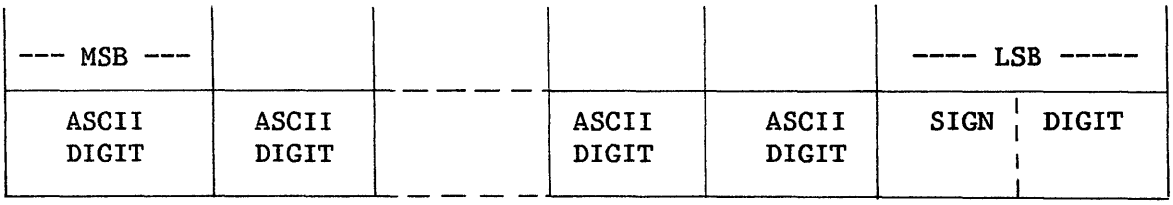
Example

"ABC1" =



Zoned Decimal

As with character data, each digit of a zoned decimal number is represented by a single byte. All digits except the one in the least significant byte (rightmost) are represented by their corresponding ASCII codes. In ASCII, the high-order four bits of the code representing each digit are referred to as the "zone" bits. The low-order four bits of each byte represent the digit itself. The least significant byte contains the sign in its four high-order bits; all zoned decimal items include a sign. (Note that in the sign portion of the byte, the value for "+" is "F", and for "-" is "D". If no sign is specified for an item, "F" is assumed.) Zoned format can be diagrammed as follows.



Before they can be used in any numeric operations, zoned decimal numbers must be converted to packed decimal format (see below). DATENTRY performs this conversion automatically when zoned items are specified as cumulative fields. Since the sign in a zoned decimal number occupies only a half-byte in memory but requires one character position on the screen, the external length of a zoned decimal item is always one greater than the internal length. The maximum length of a zoned decimal item is 15 digits plus a sign, or 14 digits plus a sign and a decimal point.

Example

"+1234" =

3	1	3	2	3	3	F	4

Packed Decimal

In packed format, each decimal digit is represented in four bits, except for the least significant (rightmost) byte of the field. The codes for the two decimal digits are placed adjacent in a byte. The sign is placed to the right of the decimal digit in the four low order bits of the least significant byte. The maximum length of a packed format item is 15 digits plus a sign. Packed format can be diagrammed as follows.

----- MSB -----						----- LSB -----	
DIGIT	DIGIT	DIGIT	DIGIT		DIGIT	DIGIT	DIGIT	SIGN

Example

"-1234" =

0	1	2	3	4	D

Sample Representations of Data Formats

Internal Format	External Value	ASCII Internal Representation	Internal Length	External Length
Character	BOOK	42 4F 4F 4B	4	4
	123	31 32 33	3	3
Zoned	123	31 32 F3	3	4
	+123	31 32 F3	3	4
	123+	31 32 F3	3	4
	123-	31 32 D3	3	4
	-.123	31 32 D3	3	5
	.123-	31 32 D3	3	5
Packed	123	12 3F	2	4
	+123	12 3F	2	4
	123+	12 3F	2	4
	123-	12 3D	2	4
	-.123	12 3D	2	5
	.123-	12 3D	2	5

APPENDIX E
RETURN CODES

All VS programs and some Procedure language statements generate return codes during execution. These return codes indicate the reasons for a program failure, or warn of conditions that could cause program failure. A successful program execution generates a return code of 0, which is transparent to the user. The return codes are listed below according to program.

Program Name	Return Code	Meaning
All compilers	1-4	Warning
	5-8	Severe error (program will not execute correctly)
	9-16	Terminal error (program will not execute)
EZFORMAT	01-04	Warning. Program will probably run.
	05-07	Severe Error. Program will not run correctly.
	08-16	Fatal Error. Object code not generated.

INDEX

WANG

Customer Comment Form

Publications Number 800-1308FM-01

Title VS FILE MANAGEMENT UTILITIES REFERENCE

Help Us Help You . . .

We've worked hard to make this document useful, readable, and technically accurate. Did we succeed? Only you can tell us! Your comments and suggestions will help us improve our technical communications. Please take a few minutes to let us know how you feel.

How did you receive this publication?

- ☐ Support or Sales Rep
- ☐ Wang Supplies Division
- ☐ From another user
- ☐ Enclosed with equipment
- ☐ Don't know
- ☐ Other _____

How did you use this Publication?

- ☐ Introduction to the subject
- ☐ Classroom text (student)
- ☐ Classroom text (teacher)
- ☐ Self-study text
- ☐ Aid to advanced knowledge
- ☐ Guide to operating instructions
- ☐ As a reference manual
- ☐ Other _____

Please rate the quality of this publication in each of the following areas.

	EXCELLENT	GOOD	FAIR	POOR	VERY POOR
Technical Accuracy — Does the system work the way the manual says it does?	☐	☐	☐	☐	☐
Readability — Is the manual easy to read and understand?	☐	☐	☐	☐	☐
Clarity — Are the instructions easy to follow?	☐	☐	☐	☐	☐
Examples — Were they helpful, realistic? Were there enough of them?	☐	☐	☐	☐	☐
Organization — Was it logical? Was it easy to find what you needed to know?	☐	☐	☐	☐	☐
Illustrations — Were they clear and useful?	☐	☐	☐	☐	☐
Physical Attractiveness — What did you think of the printing, binding, etc?	☐	☐	☐	☐	☐

Were there any terms or concepts that were not defined properly? ☐ Y ☐ N If so, what were they? _____

After reading this document do you feel that you will be able to operate the equipment/software? ☐ Yes ☐ No ☐ Yes, with practice

What errors or faults did you find in the manual? (Please include page numbers) _____

Do you have any other comments or suggestions? _____

Name _____ **Street** _____

Title _____ **City** _____

Dept/Mail Stop _____ **State/Country** _____

Company _____ **Zip Code** _____ **Telephone** _____

Thank you for your help.

Printed in U.S.A. 14-3140 3-82-5C

WANG

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 16 LOWELL, MA

POSTAGE WILL BE PAID BY ADDRESSEE

WANG LABORATORIES, INC.
CHARLES T. PEERS, JR., MAIL STOP 1369
ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851

Fold

Cut along dotted line.

WANG

ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851
TEL.(617)459-5000
TWX 710-343-6769, TELEX 94-7421

Printed in U.S.A.
800-1308-01
10-83-15M