WANG

2200

IDEAS Release 2 User Manual

2200
IDEAS™ Release 2 User Manual

1st Edition — August 1983

700-6872

WANG

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

WANG

WANG LABORATORIES, INC., ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL. (617) 459-5000, TWX 710-343-6769, Telex 94-7421

PREFACE

This version of the IDEAS Release 2 User Manual consists of two parts: a reference manual with a chapter explaining each of the components of the IDEAS utilities, and a familiarization exercise that is meant to introduce a new user to the IDEAS Release 2 utilities by guiding through a simple application program. Wherever possible, the screens used to illustrate the familiarization guide are the same as the screens used to illustrate the manual.

This manual concerns the IDEAS Release 2 Utilities. All references to IDEAS refer to this second release. The information contained in the manual only applies to the first release of IDEAS where IDEAS Release 1 is explicitly mentioned.

This manual accompanies Package Number 195-2209-3 (single-sided, single density diskettes) and Package Number 195-2209-5 (dual-sided, double density diskettes).

CONTENTS

PART I: IDEAS RELEASE 2 USER MANUAL

CHAPTER 1 OVERVIEW OF IDEAS RELEASE 2

CHAPTER 2 USING IDEAS RELEASE 2

CHAPTER 3 DATA FILE UTILITIES

CHAPTER 4 SCREEN UTILITIES

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

CONTENTS (continued)

FIGURES

FIGURES (continued)

FIGURES (continued)

TABLES

PART I
IDEAS RELEASE 2 USER MANUAL:
Components of the IDEAS Utilities

CHAPTER 1
OVERVIEW OF IDEAS RELEASE 2

IDEAS™ Release 2 (Inquiry Data Entry Access System) is an application development tool that creates BASIC-2 applications for the 2200 Series product line. IDEAS consists of a series of utilities that create and maintain data files, generate screen formats with an extensive array of field processing options, produce complex reports, and manipulate data files. The IDEAS utilities also create the data entry programs that use the IDEAS-generated data files and screens to solicit and validate operator-entered data. The variety of actions that IDEAS supports is extensive and enables you to create most applications without writing any BASIC code.

IDEAS Release 2 offers many programming options that you can implement directly through the IDEAS utilities, including hierarchical security precautions at the field, file, screen, program, and menu level. You can define subscreens and subfields, and establish any of a number of field processing actions. The IDEAS utilities offer the options of chaining screen modules and chaining program modules. The utilities provide documentation for IDEAS data files, screens, menus, and reports. Additionally, IDEAS includes several supplementary utilities that perform such actions as reconstructing files, providing conversion to and from telecommunications (TC) format, and protecting or releasing records in a file.

This chapter provides an overview of the components of the IDEAS software, describing the system utilities and the runtime components. The chapter lists the hardware requirements for developing an application using IDEAS and describes the general operating instructions for the utilities. It also presents the starting procedures for running the utilities, including explanations of the first screens you encounter when you run IDEAS.

1.1 FUNCTIONAL COMPONENTS

The IDEAS Release 2 software consists of two functional parts. The first part is the IDEAS system utilities. The utilities create data files, screen masks, data entry programs, reports, and START and menu modules. The second part of IDEAS is a set of program and data files you need to run an IDEAS-generated system. These programs, the runtime components, form the basis of an IDEAS application. They must reside on a disk in the system when you run an IDEAS-based application.

IDEAS is a registered trademark of Wang Laboratories, Inc.

1.1.1 System Utilities

The system utilities are the IDEAS programs you use when you develop an application. The Primary Program Selection menu (Figure 2-4) provides access to these utilities. Each utility creates a different component of the application; the different utilities are summarized below. Chapters 3 through 11 present complete instructions for the operation of these utilities and their options.

Data Files

The Data File utilities define, revise, and document data files. You indicate whether the file is a primary, alternate, or reference (keys only) file and define the individual fields. You control the positions of each field within a record, as well as the field's valid characters and legal decimal format. You can divide fields into subfields to any number of nesting levels. You determine the key fields for the file and the disk space reserved for that file. Once you define a record, you can still add or delete fields from the record or change the attributes defined for a field.

The maximum file size that the key structure on an IDEAS-developed system supports is 2,097,152 records in a one-volume file or 16,777,216 records in a multivolume file; of course, the actual number of records depends on the record length. A logical file can span up to eight disk platters on line; thus, the maximum file size is limited only by available memory and disk space and, in extreme cases, key size.

Each data file must have one primary key; you can associate up to 16 alternate keys with each primary data file. Each alternate key is stored in a separate file. Duplicate primary and alternate keys are allowed. You can create a reference file that contains only primary keys. The IDEAS system automatically performs all key file maintenance, regardless of file type.

Primary or alternate keys can contain up to five different fields. These fields do not have to be contiguous in the record, and any combination of five fields can form the key. You can specify the sort order for each field in the key as ascending or descending.

Screens

The Screen utilities are perhaps the most important IDEAS utilities. Through these utilities, you define data entry and retrieval screens, as well as the fields that appear on those screens. You specify all data processing performed on a field through a special "field edit" option in the Screen Definition utilities.

Using the Screen Mask utilities, you can define a screen by entering the screen mask on the CRT and defining data entry fields in the appropriate locations. When you create or modify screen masks, you can move entire lines of text in any direction on the screen mask. The utilities also allow you to use the Model 2236DE, 2236DW, 2336DE, and 2336DW terminal box graphics.

A screen that IDEAS Release 2 generates can automatically access and update up to seven different data files. An unlimited number of screens can use each data file.

IDEAS Release 2 enables you to create subscreens that can appear after you enter particular fields. A subscreen can contain fields of its own or can display explanatory text only. You can also create a HELP screen, a special type of subscreen that consists only of text and that automatically returns control to the data entry screen from which you accessed it. IDEAS Release 2 allows you to chain to another screen or even to another program. You can define chaining at any field on the screen, and you can make chaining conditional, based on a wide variety of tests.

The field processing options you define through the utilities provide the basis for the data entry programs you later define through the program generation utility. Some of these options include providing default values for fields, testing fields for a variety of conditions, establishing field attributes, or processing the fields in several ways. Additionally, you can specify up to 64 actions per field. These actions include copying fields, transmitting fields, loading programs, branching to other fields, saving records, and displaying a specified range of fields. Basically, field processing uses the field edit option of the Field Definition screen to provide the programming logic for an IDEAS-developed application.

Interactive and Batch Programs

You can generate both interactive and batch programs through the Data Entry program modules. These modules incorporate the Edit Specifications you specify during screen creation to create the Data Entry program. To create a program, specify what screen masks and/or data files the program uses. The IDEAS utilities create the program according to the field Edit Specifications of the associated screen.

A special feature of the Program Generation utilities automatically generates some standard simple programs without requiring you to specify the processing logic when you define the associated screen. This feature creates a simple data entry, random inquiry, sequential inquiry, or add/change/delete program quickly. Refer to Chapter 8 for details of this feature.

Reports

The Report Generation utility enables you to design and implement complex reports with a minimum of effort. The report definition files created by the Report/Form Printing utilities define the content and format of the report to be printed. Each report can automatically access data from up to seven data files, and can access specific fields from any data file. You can process fields in a report in the same manner as fields in a screen.

Reports can contain 11 logically-grouped levels, including headings and totaling, in addition to page totals and page numbers. The number of math functions and constants you can use for each report is virtually unlimited. You enter the text for the report mask directly onto the screen. The screen scrolls to the side to accommodate the maximum width of 158 columns for each report (in 12-pitch), and the screen scrolls vertically to allow you to define up to 99 lines.

The Report utilities include a Record Selection component. This enables you to determine that only a portion of the appropriate data will be printed, according to range tests or logical tests performed on specified key fields. The Record Selection component includes a sort specification screen that enables you to print the records of the report according to the logical sequence of up to five fields of the report. You can make Record Selection modifiable at runtime, so that the operator can specify what sections to print out and in what order.

Menus

The Application Menu Program utilities create the menus that evoke the components of an IDEAS-developed application. Each menu can call up to 17 programs or submenus. Menu entries can be IDEAS-developed programs, programs from another Wang Utilities system, or other 2200 programs.

You can include the IDEAS Release 2 Development Utilities menu as an entry on an application menu screen, allowing direct movement from an application menu to the Utilities menu. This facility allows you to modify applications during runtime. It also facilitates testing of applications during the development cycle.

START Programs: Application Initialization

The START Program utility creates the application initialization program for an IDEAS-developed application. This START module initializes system addresses and operating parameters and opens specified files. The START program loads a specified program. Typically, the START program loads a main menu from which an operator can select the different components of the application. You can chain Application Initialization programs together, allowing one application menu to contain several other applications subsystems.

Supplementary Utilities

Several supplementary utilities increase the power of IDEAS-developed software systems. The File Recovery utility reconstructs damaged data files from information saved on disk. A utility to convert IDEAS files to standard telecommunications format allows you to use any standard Wang TC utilities and batch emulators; this utility also converts these TC files back to IDEAS format. A set of utilities converts IDEAS Release 1 files, including data files, screens, and reports, to IDEAS Release 2 files. IDEAS Release 2 also provides utilities that check file status, protect all records in a file, release all records in a file, and expand the size of an IDEAS 2 file, saving the data in the file. Additional utilities maintain the Edit Specification file and the CPU system file.

Security Features

IDEAS Release 2 offers an extensive security system. Each user of the IDEAS utilities or IDEAS-generated programs is assigned a User ID, one of 16 hierarchical user classes (0 through 9 and A through F, with A greater than 9), and, optionally, a password. You can restrict access to programs, screens, and data files by means of User ID or user class. In addition, you can protect programs, screens and data files with password security. IDEAS-generated program menus do not display programs that are not available to a user of a particular class.

Any user of the highest user class (User Class F) who knows the security file password can maintain the list of User IDs, user classes, and user passwords. You should modify the password for the security file immediately after installation, using the system menu module. For instructions on how to do this, refer to Subsection 2.3.3.

1.1.2 Runtime Components

Running an IDEAS-developed application requires that the IDEAS runtime components be resident on a disk in the 2200 System. These files perform the various tasks of the application. The Install an Application utility, described in Section 11.10, transfers the files needed to run an IDEAS application. Appendix F contains a listing of these files.

Two major runtime components are HIKAM (Hashed Index Keyed Access Method) and the System-Resident Subroutines.

HIKAM

HIKAM is a unique access method that handles all disk I/O and storage automatically. This method allows the system to access data readily and provides rapid storage and retrieval of data records. HIKAM provides excellent performance in a random storage/retrieval environment. HIKAM uses both hashing and indexing techniques, resulting in a system with optimum performance.

System-Resident Subroutines

The System-Resident Subroutines are subroutine calls that form the basis of all phases of an IDEAS-developed application. For example, System-Resident Subroutines perform all file access, key file maintenance, data packing and unpacking, and actual disk operations automatically. The development utilities and the generated applications use the same set of subroutines, allowing easy movement between the development stage of an application and the running of an application. You can run the System-Resident Subroutines in a global or local partition.

1.2 ADDITIONS TO AN IDEAS-GENERATED PROGRAM

The code generated by IDEAS Release 2 usually requires no modification. The various utilities, particularly the Screen utilities and Report utilities, provide an extensive variety of processing options and program chaining. However, when you require specialized field editing logic or processing logic beyond that already supported by IDEAS Release 2, you can use the user exit facility.

User exits provide a convenient way of supplying specialized logic to IDEAS-generated programs while still taking advantage of the self-documentation and easy maintenance of IDEAS. It is easy to insert subroutine calls to IDEAS subroutines as well as user-written subroutines at the appropriate section of the code. User exits are defined in Appendix C and in Section 5.3 under the heading "User-Supplied Process."

1.3 HARDWARE REQUIREMENTS

Table 1-1 presents the hardware requirements for developing or running an application using IDEAS Release 2.

Table 1-1. Hardware Requirements For IDEAS Release 2

Category	Requirement
CPU	Any VP-based CPU in the 2200 line (i.e., 2200VP, 2200SVP, 2200MVP, 2200LVP) using Release 2.2 or greater of the BASIC-2 operating system.
Terminal	The Model 2236DE, Model 2236DW, Model 2336DE, and Model 2336DW terminals are standard. Although IDEAS Release 2 operates on earlier models, such as the Model 2236D and Model 2236, use with these terminals may be confusing to the operator on a screen that does not support box graphics and field attributes.
Disk	Any Wang disk drive supported by the above CPUs operating under BASIC-2. The size of the IDEAS files requires at least one hard disk to run the development utilities, although a simple application could require only a DSDD diskette.
Printers	All standard Wang 2200 printers
Memory	2200VP Operating System - 28K 2200MVP Operating System - Two modes of operation exist, local and global. Application Development Local mode: a single foreground partition of 28K. This configuration is compatible with other Wang software packages, such as 2200WP, Datamerge, and MAILWAY[R]. Global mode: 14K of background memory and 17.5K of foreground memory. Refer to Section 2.3 for information on background/foreground memory trade-off. Running an IDEAS-developed application 28K of local memory or 17.5K of local memory and from 12.25K to 14K of global memory. A particularly large application may require more memory and, conversely, a small application may require less memory.

MAILWAY is a registered trademark of Wang Laboratories, Inc.

1.4 STORAGE CONSIDERATIONS

Determine the disk configuration from which you will run the developed application before you begin application development. Running an IDEAS-developed application requires both the runtime components and the appropriate IDEAS-generated files. The runtime support modules require approximately 700 sectors; Table 1-2 summarizes the sizes of the application files. The runtime support modules need not be on the same disk as the application program, though it may prove more convenient if they are. If you will use a hard disk for the application, you should move the runtime support modules to that disk; for information on how to install an IDEAS-developed application, refer to Section 11.11.

An IDEAS-based system makes more entries to a disk catalog index than might appear during application development. This is because data files, report files, and batch program files create a companion file, called a definition file, containing IDEAS system control information. Table 1-2 lists the types of files created by each utility, the number of sectors contained within files, and the number of entries made to the catalog.

Table 1-2. Storage Considerations

File Type	Definition File	Program or Data File	Index Entries
Primary Data	18 sector	N* sector data	2 per data file
Alternate Key	4 sector	N* sector data	2 per alternate key file
Report and Batch Programs	N* sector	N* sector program	2 per report
START	None	11 sector program	1
Menu	None	8 sector program	1 per menu
Data Entry	None	N* sector program	1 per data entry program
Screen	None	27 sector data (Less for an installed application)	1 per screen

* The number of sectors used is variable. The number of sectors required for a data file is displayed during file creation.

IDEAS names the definition files automatically. The definition file name is a lowercase version of the name entered in the utility. Thus, each file name entered must have at least one uppercase letter to distinguish it from the definition file.

Table 1-3 provides a partial list of the contents of each definition file.

Table 1-3. Contents of Definition Files

File Type	Contents of Definition File
Data File	The definition file contains information about on the keys, character compression, the field names and attributes, and the disk address.
Report	The definition file contains information about the report format, such as report mask definitions, level breaks, and grouping levels.
Screen	The Screen file can be considered a definition file since it defines the screen creates and drives the interactive programs. It contains the screen mask used at runtime to display the screen, as well as a list of field names, locations, and characteristics.

IDEAS creates some files that you should take into account when determining the space necessary for IDEAS development: an Edit Specification file (IDS2fs00), and a security file (IDS2f005). You determine the size of the Edit Specification file and security file when installing the IDEAS utilities, as described in Section 1.5. You do not need the Edit Specification file to run a developed application, although you do need the security file.

Besides the Edit Specification file and security file, IDEAS creates several additional files, which are summarized in Table 1-4.

In short, you should allow sufficient space in the disk catalog and catalog index to account for the necessary IDEAS development files, runtime files, work files, and IDEAS-generated application and definition files. IDEAS offers a multivolume file option that allows you to spread a data file out over several disks.

On a 2200 system, you should avoid catalog index sizes that are multiples of three; the nature of the hashing algorithm makes catalogs of such sizes less efficient. Because IDEAS uses a large number of program files, it is especially important to choose an index size (the LS parameter on the SCRATCH DISK command) that enables efficient use of the catalog index. Generally, prime numbers provide the best performance.

Table 1-4. Additional Files Created by IDEAS

File Name	File Size	When Created	Purpose
IDS2fI00	3 sectors	Runtime	Saves system parameters.
IDS2f001	3 sectors	Development and runtime	CPU master file: notes which CPUs are accessing the platter.
IDS2fs0#	166 sectors per CPU	Development and Runtime	Stores device addresses and @TSTFLDs. The variable # indicates CPU number.
IDS2PX##	10 sectors per station*	Report Execution	Record selection specification. The variable ## indicates station number.*
IDS2wS##	186 sectors per station*	Runtime	Work file. The variable ## indicates station number.*

* The station number is unique for each partition of each CPU using IDEAS. If there is only one CPU, the station number is the partition number. If there is more than one CPU, the station number is the partition number plus a multiple of sixteen, depending on the number of CPUs.

1.5 INSTALLING THE IDEAS SOFTWARE

The IDEAS software includes the stand-alone utility IDS2MOVE. This utility allows you to install the modules needed to develop an IDEAS application. The development system consists of approximately 250 programs and 100 screens, plus several system files that must be allocated and initialized the first time you install the developement system.

Running the installation utility requires a single partition of 28K or larger. To run the install utility, select the address where the first diskette resides (touch RESET, and then type SELECT DISK xxx and touch RETURN). Then LOAD and RUN "IDS2MOVE", which requests a source disk address for the software (the address just selected), requests a destination disk address, and asks whether to verify after the copy.

After you enter the above three items, the software is copied by using ISS reference files that are also supplied with the IDEAS software. The number of files being copied is displayed, along with their names. As each diskette is completed, a prompt appears instructing you to mount the next diskette.

If the development system was installed on the destination platter previously, then an IDEAS security file, CPU ID master file, and Edit Specification file already exist on that platter, and processing for the install utility ends. If the IDEAS utilities are being installed for the first time, the following processing occurs.

The CPU File Maintenance screen appears, describing the contents of the IDEAS CPU master file (IDS2f001) that contains the current valid CPU ID numbers. The CPU ID is the number returned by the BASIC-2 function PRINT #ID, for which Operating System Release 2.2 or greater is required. It is a five-digit number if the CPU has the necessary hardware; otherwise, it is zero. This file is necessary because IDEAS allows up to six different CPUs to share a single disk drive (by multiplexing). IDEAS uses this master file to determine which CPU is the first one in an installation, which is the second, and so forth. This information, in turn, is used for a variety of purposes, including identifying the user holding (protecting) a data record, controlling access to work files, and providing unique device assignments (in the development system) to each developer. After an IDEAS system is installed, you can change or delete CPU ID numbers through the CPU System File Maintenance utility, described in Section 11.14.

In most cases, there is no need to supply any information for the CPU File Maintenance screen. However, if the CPU you are using for installation does not have its ID number in the IDEAS CPU master file, a prompt appears asking whether to add the CPU number to the CPU master file or to replace an existing CPU ID with it. Refer to Appendix A for information about CPU multiplexing.

You must now allocate space for the system security file IDS2f005 and for the Edit Specification file IDS2fs00. Allocating the IDEAS files here is similar to allocating and initializing any IDEAS data file, as described briefly below and in more detail in Section 3.2.

The IDEAS system security file contains a user ID, a name, a user class, and (optionally) a password for each person using IDEAS. You can later add valid users to the file with the Security option of the IDEAS main menu. (Refer to Subsection 2.3.3.) The disk space allocation screen asks for the number of records desired for this file and displays the number of disk sectors these records will require. You can experiment with different numbers of records to see how much disk space they will use, before you finally accept the screen. Each user requires one record, and four records fit into one disk sector. Accepting the allocation screen initializes the security file.

The allocation screen reappears for the Edit Specification file (IDS2fs00). This file, used to define field processing and validation when creating screens and reports, requires one sector per record, and a typical edit requires three records. Most applications require between 100 edits (330 disk sectors) and 500 edits (1650 disk sectors), assuming that the developer archives screens and reports after completing the design process and generating the accompanying programs. Refer to Chapter 4 for further discussion of archiving screens and Chapter 5 for a discussion of Edit Specifications.

You can later expand the IDEAS system files created through the Install utility if they are not large enough. The IDEAS system files cannot be multivolume.

You should keep the IDEAS diskettes and the Install utility in case it becomes necessary to reinstall the IDEAS system.

The IDEAS system utilities include a set of library screens you can use to retrieve some common processing operations. You should install these screens and familiarize yourself with them before beginning any program development. Chapter 5 presents information about these screens and their installation.

1.6 INSTALLING AN IDEAS-DEVELOPED APPLICATION

Running an IDEAS-developed application requires the IDEAS runtime utilities as well as the program and data files generated for the application. The IDEAS package includes a utility to copy these files onto a separate platter or series of platters. This utility is described in Section 11.10.

CHAPTER 2
USING IDEAS RELEASE 2

2.1 PLANNING AN APPLICATION

Before you run any of the IDEAS Release 2 utilities, you should decide what is required of the application you are going to develop. You should determine the data that needs to be stored, how that data is organized into different data files, and what fields are to be used as primary and alternate keys. Most importantly, you should determine the logical operations for the data entry program. You should decide upon the following:

Data Files

Are single or multiple volume files to be used?
How many records will be required?
What type of primary key is most desirable?
What types of alternate keys are most desirable?
How should the data files be organized?
When and where should a field be divided into subfields?

Reports

What kinds of reports are needed?
What field or fields should determine the order of the reports?

Screens

How should the screens be organized?
What subscreens and HELP screens should there be?
What processing should occur for each screen?

General Concerns

How should security be implemented throughout the application?
How should the Version, Application, and Function fields be used to organize the application? (Refer to Section 2.2.)

You should plan the IDEAS-developed program before running the Screen utilities, because they define the programming logic. Chapter 5 discusses the programming capabilities of the utilities. You should read this manual in its entirety and create the application described in the familiarization exercise before you begin your first application.

When developing an application, you must first run the Data File utilities. You need not run the remaining utilities in any particular order, but you may find the following sequence convenient:

1. Data Files (always first)
2. Screens
3. Programs (Interactive)
4. Reports and/or Batch programs
5. Menus
6. START programs

2.2 OPERATING FEATURES

There are certain operating features of IDEAS that you should keep in mind while running the utilities. These characteristics hold true throughout the system utilities.

2.2.1 General Instructions

IDEAS is a menu-driven system that employs a hierarchy of menus; there is one main menu screen from which you can evoke the main menu screen for each component of the IDEAS utilities. Functions associated with FN 01 through FN 05 are unique to each component of IDEAS; functions associated with FN 06 through FN 15, which provide access to the major IDEAS components, remain constant throughout all menus. In other words, you need not return to the main menu to access other system utilities. IDEAS is screen-oriented; you must enter information or select options from screens throughout the system.

All of the fields on the system utilities screens terminate (cursor moves to the next field) automatically when full. Of course, if a field entry is less than the full size of the field, you must touch the **RETURN** key to make that entry. This characteristic applies to screens as well; when the last field on a screen is full, the system automatically proceeds to the next screen. Additionally, using the Screen Mask utilities, you can define your application screens to incorporate automatic termination of full fields.

Throughout the IDEAS utilities, the screen prompts indicate that you should touch the **EXECUTE** key to accept a screen. This instruction applies to a Model 2236DW or 2336DW terminal. On a Model 2236DE or 2336DE terminal, touching the **RUN** key accepts a screen. On both types of terminals, you can also use the **RETURN** key to accept most screens. On a Model 2236DW or 2336DW terminal, the cursor keys (N, S, E, and W arrows) control the cursor when you are defining a screen or a report mask.

You can control cursor movement on most system utilities screens with FN 04, 07, and 11 through 14, the cursor movement keys along the top of the terminal keyboard. These keys backspace to previous fields or skip several fields at a time when you are editing a screen; FN 11 is sometimes referred to as the skip-ahead key. On a 2236DW or 2336DW, the cursor movement keys are equivalent to FN keys 04, 07, 12, and 13.

When you wish to edit a field you have entered already, move the cursor to any character in that field and touch the EDIT key. This underlines the field and places you in EDIT mode, allowing you to edit portions of a field. In EDIT mode, the cursor keys move the cursor within the field, and typing any letter overstrikes the current character in the field. When not in EDIT mode, striking any character in the first position of a field blanks that field and allows you to reenter the field. Throughout the utilities, you can use FN 31 as a CANCEL key, returning control to the previous screen. When the screen prompts indicate that you can cancel, FN 31 fulfills this purpose.

The IDEAS software provides its own set of prompts and error messages that appear at the bottom line of the screen when appropriate. Appendix H lists these messages and Chapter 5 provides instructions for disabling and testing these messages in an application.

If a 2200 system error outside the realm of IDEAS occurs while you are developing or running an IDEAS application, the system does not crash. Instead, a message appears at the bottom of the screen indicating the BASIC-2 error code, the IDEAS module in which the error occurred, and the line of code that yielded the error. Pressing EXECUTE at this point returns you to the most recently encountered menu.

2.2.2 Version, Application, Function

The second screen of each of the major utilities presents the fields Version, Application, and Function. These fields establish groupings of IDEAS files. They act as selection parameters for all IDEAS-defined batch processing, including batch documentation, screen archiving, and screen update.

When creating or editing the specifications for a data file, screen, data entry program, menu, or start program, you can fill in the Version, Application, and Function fields with an alphanumeric code or abbreviation identifying the group to which the field belongs. IDEAS imposes no defined hierarchy of usage on the three fields. They are interchangeable, differentiated only by their varying lengths. Your own conventions supply any hierarchy or relationship among them.

You can use the Version, Application, and Function fields to install a developed application, as described in Section 11.10. Thus, it is important to use the fields to identify each separate application.

You can use Version, Application, and Function in batch documentation to select files and programs with identical parameters. The selected files and programs cannot be password-protected and must be available to the current user class and ID.

2.2.3 Security

Security features for IDEAS files appear in two phases: during application development, and at runtime. Security plays a part in all major components of IDEAS development, which are as follows:

- Data files - at both file level and field level
- Screens
- Data entry programs
- Batch programs
- Menus
- Start programs
- Reports

All development utilities provide security for the system components through user class, user ID, and password. If you specify a user class, a user of a class lower than the specified class cannot edit or document the component. If you specify a user ID, no one other than the specified user can enter that component of the application. If you specify a password, the user must provide the right password. Any or all of these restrictions can bar the unqualified user from access to or knowledge of the design of a system. Subsection 2.3.1 describes entry to the system, and Subsection 2.3.3 describes security administration.

When creating or editing data entry programs, you can protect them from listing, examination, or modification by choosing the Storage Security option from the development screen.

2.2.4 Compression of IDEAS Files

> **WARNING**
>
> There are 13 IDEAS subroutines that you should not compress using the Wang Integrated Support System (ISS) software package; Appendix F identifies these subroutines. You can compress IDEAS-generated code, but compressed code cannot be modified.

2.3 DAILY INITIALIZATION

As described in Section 1.3, you can run IDEAS in local or global mode. In local mode, the major IDEAS subroutine modules load into the first thousand lines of memory in the foreground partition. This is the mode used automatically with a VP operating system. In global mode, a global subroutine running in a separate partition contains the major IDEAS subroutine modules, and global subroutine calls issue from the foreground partition when required. This mode can operate only on MVP or LVP hardware.

The decision to run IDEAS in local mode or global mode depends upon the current required configuration, which in turn depends on the applications to be run. By running IDEAS in local mode, once person can run IDEAS while another runs 2200WP in separate 28K partitions within the same bank. Running IDEAS in global mode allows two partitions in a single bank to run IDEAS and share the global subroutines. This might prove useful when two users wish to run large applications that require excess memory; a 14K global and two 21K foreground partitions provide each user with the equivalent of 34K of memory.

Before running IDEAS in global mode, you must configure the system properly, with adequate memory in both the foreground and background partitions. The size of the foreground partition can vary, depending on the size of the application being developed or run. Usually, 17.5K is sufficient memory for the foreground partition, but a large application requires more. The background partition can vary from 12.25 to 14K; the size of the background partition determines which subroutines it contains. Table 2-1 lists which subroutine modules are missing from a background partition of less than 14K.

If a missing subroutine module is needed for a selected processing, the appropriate module is paged into the local partition, where it requires about 1K of memory. Thus, a small background partition could require a larger foreground partition. All of the above subroutines share the approximately 1K of extra memory. For example, if date validation has been loaded and then sequential processing is needed, date validation is cleared from foreground to allow room for sequential logic.

Table 2-1. Subroutines Paged into Local Memory

Size of Background Partition	Subroutines Paged into Local Memory
12.25 to 13K	Date validation and conversion, sequential processing
13.25K to 13.75K	Date validation and conversion
14K or more	All modules present in global memory

When either an IDEAS-developed application or the IDEAS utilities are loaded and run, they perform the following steps:

1. The utilities check whether a set of global subroutines exists in a separate partition in the current bank. If so, IDEAS selects global mode.

2. If the global mode is missing, the utilities check whether the current partition contains at least 28K of memory. If so, IDEAS selects local mode.

3. If both Tests 1 and 2 fail, execution stops.

Therefore, to run IDEAS Release 2 in local mode, simply LOAD and RUN "IDEAS2" when in a partition of at least 28K after ensuring that the global subroutines are not contained in another partition in the same bank. To run IDEAS Release 2 in global mode, LOAD and RUN "IDS2GLOB" into what will be the background partition, which should be in the same bank as the desired foreground partition. This program releases its terminal, putting you into another partition within the same bank. If this is the partition from which you will run the utilities, LOAD and RUN "IDEAS2".

Running IDEAS evokes the IDEAS Disk Address Selection Screen, Figure 2-1.

```
I.D.E.A.S.          Inquiry, Data Entry, and Access System          Release 2.1

          This module requires that you specify the disk address
          where the I.D.E.A.S system utilities reside.  The disk
          addresses supported by I.D.E.A.S. are listed below.

          Please choose the appropriate address for your system
          from those listed and touch RETURN, or just touch the
          RETURN key to accept the default address as shown.

          The disk address you choose (if valid) will become the
          new default address for this module on this terminal.

        ------< Valid disk address supported by I.D.E.A.S. >------
        310  320  330  B10  B20  B30  350  360  370  B50  B60  B70
        D10  D11  D12  D13  D14  D15  D50  D51  D52  D53  D54  D55
        D20  D21  D22  D23  D24  D25  D60  D61  D62  D63  D64  D65
        D30  D31  D32  D33  D34  D35  D70  D71  D72  D73  D74  D75

        Please enter I.D.E.A.S. system utilities disk address. D23
```

Figure 2-1. IDEAS Disk Address Selection Screen

If the default address shown in the bottom right corner of the screen is the correct address, touch EXECUTE. If it is not, enter the appropriate disk address from those listed on the screen, and then touch EXECUTE. If an error message appears, check that the disk is properly mounted at that address.

2.3.1 Security Access Screen

After you accept the Disk Address Selection screen, the Security Access screen, Figure 2-2, appears.

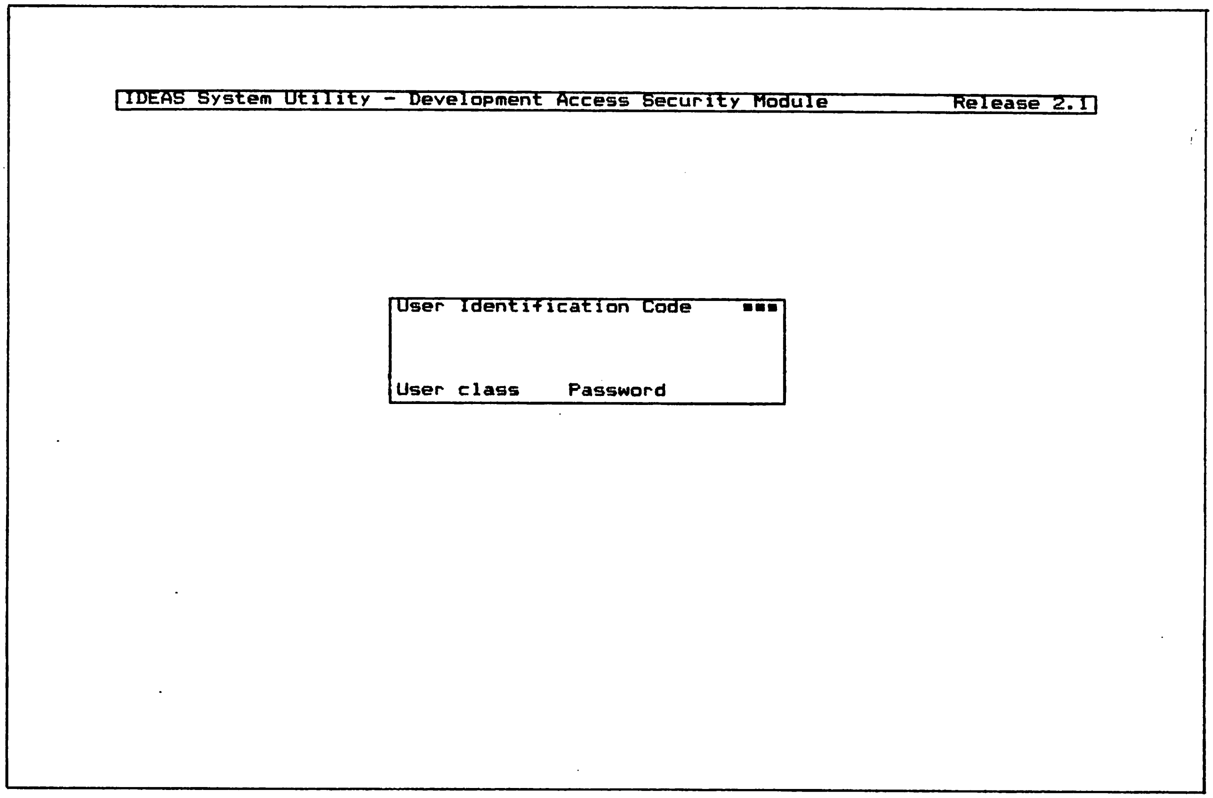


Figure 2-2. Security Access Screen

The Security Access file, defined at installation, resides on the IDEAS software. This file contains the valid user IDs and associated user names, classes, and passwords. When the Security Access screen appears, you must enter a user ID. If you are included in the security file, the system displays your name and user class. If a password is specified for your entry to the system, you cannot access the system until you have given your password.

As with most other fields in IDEAS, when the user ID, password, and other security fields are full, the cursor automatically skips to the next field. In other words, you need not touch RETURN after filling the last blank in the field. A mistaken RETURN may bypass a field requiring your attention; if this occurs, you should correct the error by returning to the field with FN 13 (the backspace arrow).

The first time you run the IDEAS utilities after installation (or whenever the security file is empty), you do not need a user ID or password. The first person to run the utilities enters directly into the security file management utility, allowing additional users to enter the system. Your first entry into the security file should be the security administrator; you must assign the security administrator a user class of F. The first person to run the utilities, or the security administrator, or any Class F user can assign other users their IDs, class, and password. Refer to Subsection 2.3.3 for information on security administration.

2.3.2 System Date Module

The System Date Screen, Figure 2-3, appears after you enter a valid user ID and password.

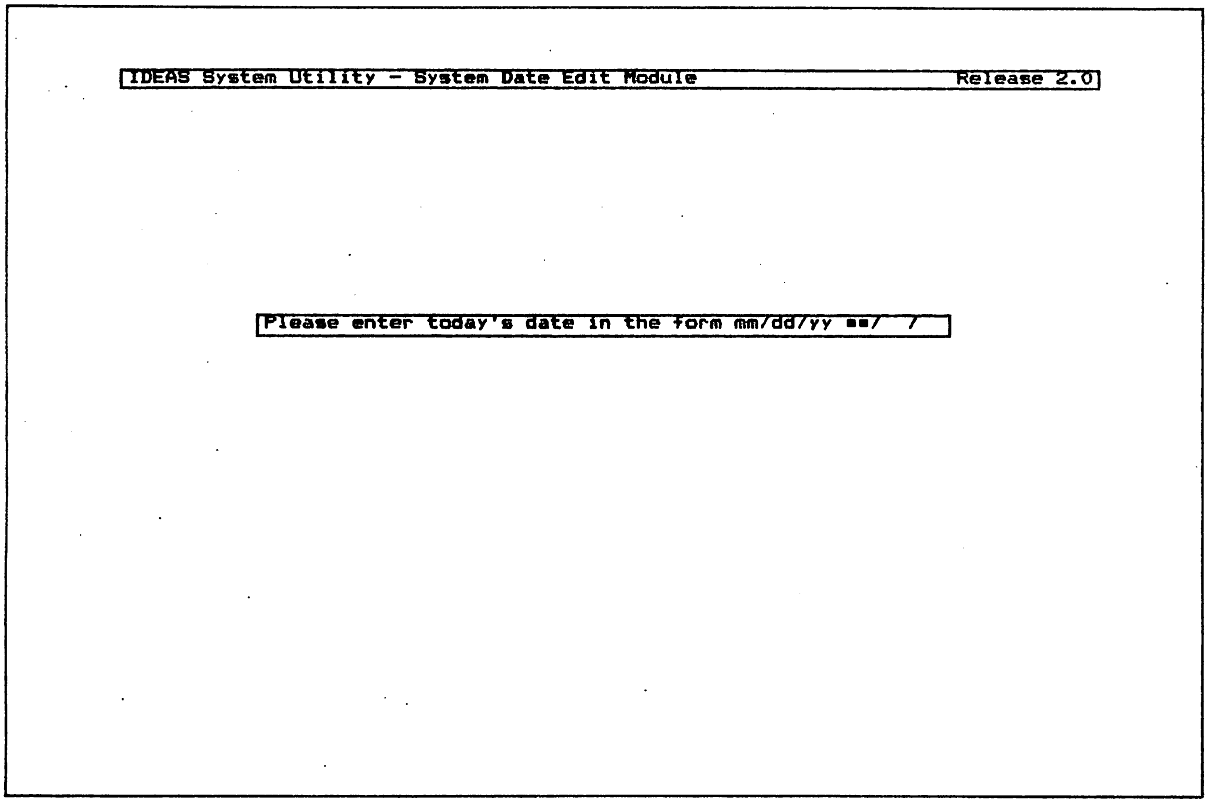


Figure 2-3. System Date Screen

The cursor appears at the beginning of the six-character date field in the center of the screen. Enter a six-digit date in MMDDYY format.

When running in local mode, you must enter the date each time the IDEAS utilities are loaded. When running in global mode, you must enter the date when the global subroutine is first loaded; subsequently, if the global subroutine is not cleared, this screen is bypassed.

If the date is valid, the system accepts it and asks if the date is correct. Touch EXECUTE to accept the date or EDIT to change the date. If the date is invalid, an error message appears on the bottom line of the screen. The cursor returns to the date field, and you must input the date again.

2.3.3 Primary Program Selection Menu

After you accept the date, the system proceeds to the Primary Program Selection menu, Figure 2-4.

```
IDEAS System Utilities - Primary Program Selection Menu Module      Release 2.1

                          FN Program Module
                        ■ 00 Security
                        • 01 System date
                        • 02 Device addresses
                        • 03 Manage system peripherals
                        • 04 New user's ID
                             -- Other Functions --
                        • 06 Data files
                        • 07 Screens
                        • 08 Programs (interactive)
                        • 09 Programs (batch)
                        • 11 Reports
                        • 12 Menus
                        • 13 START programs
                        • 14 Other utilities
                        • 15 Applications

SEP 27 83                                                          Term #  2
Mem 36.00                   Global Selected                        Sta. # 20

Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 2-4. Primary Program Selection Menu

This screen allows you to proceed to any of the functional components of IDEAS by touching the appropriate function key. The major utilities of the IDEAS software, which are evoked by touching FN 06 through FN 14, are described in detail starting with Chapter 3. Explanations of the remaining selections of the main menu follow. Note that the Primary Program Selection menu indicates whether the system is operating in local or global mode.

FN 00 -- SECURITY

This option appears on the screen only when the user entered on the Security Access Screen (Figure 2-2) or from the New User ID option (see below) is class F. This option allows a security administrator (anyone with User Class F) to add new users to the system, edit the information on current users, and delete current users.

When you choose this option, a prompt appears at the bottom of the screen requiring you to enter a password to enter the utility. The password is initially PASSWORD. Any class F user can change the password by revising the master menu (IDS2M001) using the IDEAS Menu Revision utility, and then protecting the master menu from further modification by indicating on the menu definition screen that a password or user ID is necessary to edit the menu.

When you enter the password, the screen shown in Figure 2-5 appears.

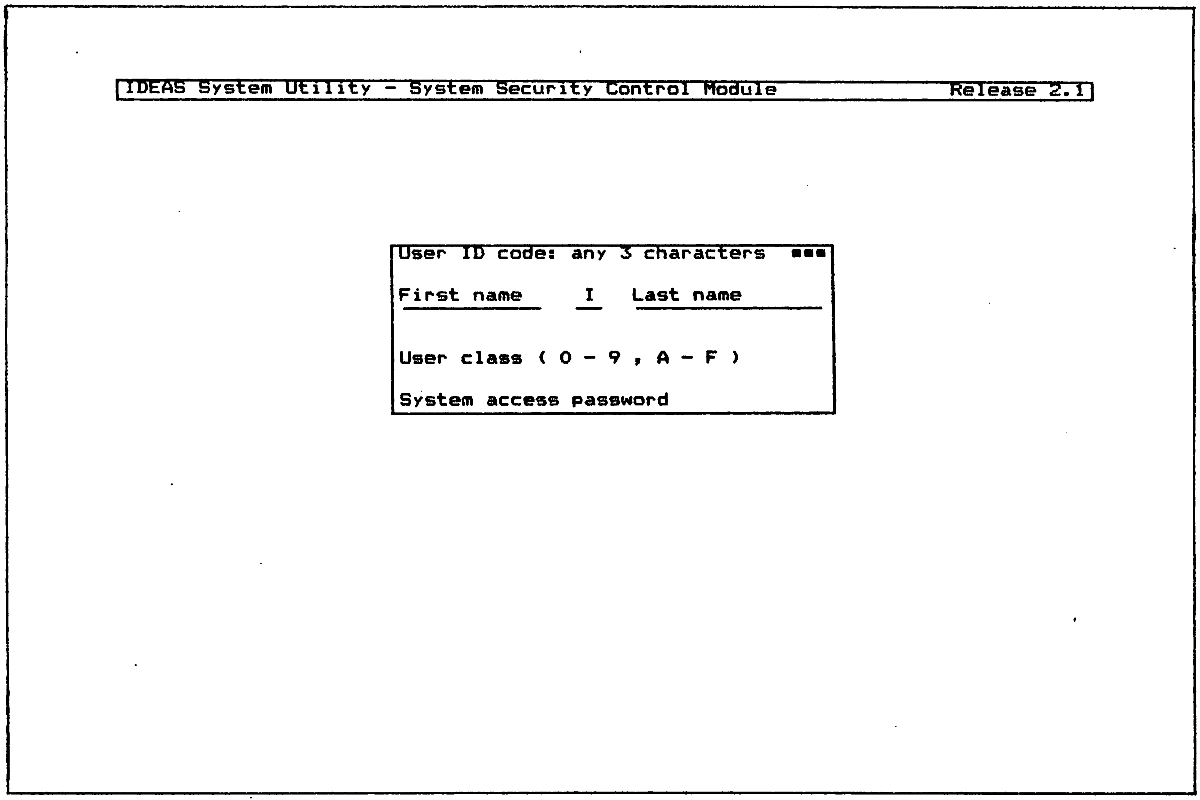


Figure 2-5. New User Screen

Enter a three-letter user ID. If this is the user ID of a new user, the system allows you to enter a name, user class, and password. If the user ID is already on the system, the system recalls the name, user class, and password and a prompt appears at the bottom of the screen allowing you to edit or delete the information before returning to the Primary Program Selection menu. IDs are uniquely assigned to an individual or group of users. Classes and passwords may be assigned identically over an entire group of IDs or to a single ID. Any group of characters can constitute a user ID.

You can specify that no password be required for entry to the system, so that when the user encounters the Security Access screen, a RETURN is sufficient to move to the next screen.

A password is not displayed on the screen as you enter it in the blanks, but the cursor advances within the password field to indicate the number of characters struck. You can edit the password by backspacing and striking over any incorrect characters.

If you should delete all Class F users from the system or lose all Class F passwords, you must run the utility IDS2PUSC, which is provided with the IDEAS utilities. This utility repairs a damaged security file by clearing out all current information and creating the file anew; it should therefore be used with great caution. The next time you run the utilities, the Security Access screen (Figure 2-2) allows you to enter a new security administrator, as described in Subsection 2.3.1.

You can restrict programs, screens, and data files developed using IDEAS Release 2 to a minimum user class. You can also divide programs, screens, and files according to edit privilege and access privilege; for example, you can define a particular screen so that anyone with a user class of 0 and above can view the screen as part of an application, but only Class A and above can edit it. Throughout the IDEAS development and runtime programs, menus display only those items permitted to your user class. For example, Security appears on the Primary Program Selection menu (Figure 2-4) for User Class F only.

FN 01 -- SYSTEM DATE

The date that you enter when you load the IDEAS utilities (whether for development or execution of an IDEAS application) remains as the current system date throughout the development and execution of an IDEAS application. You can use this date as a variable on a report or on a screen, and the system notes the date of the most recent revision of an IDEAS file. You can touch FN 01 to change the currently entered date without canceling out of IDEAS and reloading the utilities.

FN 02 -- DEVICE ADDRESSES

You must verify the Peripheral Device Selection Screen, Figure 2-6, before beginning any application development. Enter the device addresses for the various components of the application being developed at this session on this screen.

```
IDEAS System Utility - Peripheral Device Selection Module          Release 2.1

                    Device # 01 / 216    Printer
                    Device # 02 / D23    IDEAS system utilities
                    Device # 03 / D21    Screens & program control files
                    Device # 04 / D21    Application program files
                    Device # 05 / D21    Data file control files
                    Device # 06 /        Telecommunications controller
                    Device # 07 / D21    Application data files
                    Device # 08 / D21    Application data files
                    Device # 09 / D21    Application data files
                    Device # 10 / D21    Application data files
                    Device # 11 / D21    Application data files
                    Device # 12 / D21    Application data files
                    Device # 13 / D21    Application data files
                    Device # 14 / D21    Application data files
                    Device # 15 / D21    Application data files or other

Touch FN key corresponding to device to be changed, EXECUTE to accept as shown ■
```

Figure 2-6. Peripheral Device Selection Screen

If you need to change any of the addresses on the screen, touch the function key corresponding to the device number and enter the correct address. The cursor moves to the first position in the address for the specified device. Enter the desired device address; three characters are required.

Separate device addresses exist for data file control files (Device 05) and Application data files (Devices 07 through 15). This is because for each data file created, whether single or multivolume, the system creates a companion control file that contains information about the file. This file may be stored in a separate location from the Application data file(s).

Similarly, separate device addresses exist for report program control files (Device 03) and for report program files (Device 04). Device 03 also determines the location of screen files, and Device 04 determines the location of all IDEAS-generated programs, including report, START, menu, interactive, and batch programs.

Device Addresses 08 through 14 are device addresses that are valid addresses for additional volumes of multivolume data files, and Device Address 15 is a valid address for an additional data file volume or an archive address for screens and reports. Only addresses that appear on the Peripheral Device Selection screen can be used for archiving. If you need to archive to a device whose address is not listed among addresses 01 through 07, enter the address you need as Device 15.

Each station on the system has a unique set of device addresses. Thus, each station can operate with different sets of device specifications.

You can include the Device Address option as part of the application that you are developing. Refer to Section 11.1 for further information.

FN 03 -- MANAGE SYSTEM PERIPHERALS

This option allows you to specify default values for the type of printer, the pitch, and the font for up to four printers. The default values specified here are used when an IDEAS-defined report is run during an IDEAS application. The sequence of screens that appears to an operator running a report is described in Section 6.10.

FN 04 -- NEW USER'S ID

Touching FN 04 evokes the Security Access screen, which allows you to change the currently selected user without restarting the system, and returns control to the Primary Program Selection screen. FN 04 serves as a log-off procedure. As soon as you invoke it, the previous user ID is no longer actively logged on to the system. A privileged user of any IDEAS development or application material should touch FN 04 when leaving the terminal in order to keep privileged materials inaccessible to lower-clearance personnel. Subsequent user entry is identical to entry at IDEAS start-up. You may put this function on an application menu by including the program IDS2P005 as a menu entry on an application menu and making sure that the security file, IDS2f005, is part of the application. Refer to Chapter 7 for information on defining application menus.

Changing the currently selected user might allow editing privileges or access to files only available to a different user ID or class. The system notes the ID of the most recent user to edit each IDEAS file; you can touch FN 04 to enter the user ID you wish to associate with the current revisions. This utility does not enable you to add new users to the system; you must use the Security utility (see above) for that.

FN 15 -- APPLICATIONS

After an application is developed, you can run the application by entering LOAD RUN followed by the name of the Application Initialization program (the START program), or you can touch FN 15 and then enter the name of the program. To facilitate the testing, debugging, and development cycle, you can make it easy to return to the development system by putting IDEAS2 on the application menu.

CHAPTER 3
DATA FILE UTILITIES

3.1 OVERVIEW

The IDEAS Release 2 Data File utilities define, initialize, and document all IDEAS data files. The utilities can also alter the parameters describing these files and reinitialize the data files after changes are made.

The Data File utilities create three different types of data files: primary data files, alternate key files, and reference files. A primary data file is a file of data records with a defined key used to locate a record in the file. An alternate key file is a file consisting of keys and pointers that provides a second means of keyed access to a primary data file; you cannot define an alternate key file without first defining the primary file that it references. A reference file is a file containing only keys; in a reference file, the record consists solely of its key field(s). Section 3.2 explains further subdivisions of the three types of data files. For each data file, the IDEAS utilities automatically create a control file that describes the structure and contents of the data file.

This chapter describes how to define and allocate data files and their fields. For information on reading from data files and saving records to data files, as well as such topics as maintaining exclusive control over data files, refer to Chapter 5.

> **NOTE**
>
> You should back up all data files to provide security against hardware errors or operator errors.

When you choose the data file option from the Primary Program Selection Screen, the Data File Utility screen (Figure 3-1) appears.

```
IDEAS Data File Editor - Utility Selection Menu Module              Release 2.1

                    FN Program Module
                  ■ 00 Create
                  • 01 Revise
                  • 02 Document
                  • 03 Initialize
                  • 04 Document (Batch)
                  • 05 Copy a data file control file
                       -- Other functions --
                  • 07 Screens
                  • 08 Programs (interactive)
                  • 09 Programs (batch)
                  • 11 Reports
                  • 12 Menus
                  • 13 START programs
                  • 14 Other utilities
                  • 15 Applications
                  • 31 Main IDEAS menu

SEP 27 83                                                           Term #  2
Mem 36.00               Global Selected                             Sta. # 20

Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 3-1. Data File Utilities

The first six options on the Data File Utility screen (FN 00 through FN 05) are summarized below. FN 00 through FN 04 and their associated screens are described in detail following the summary.

FN 00 -- CREATE

The file creation module allows you to create one of the three different kinds of data files described above and to define the parameters of the file. The system presents a sequence of screens that ask you to enter the file type, define the fields of the record, choose the key field or fields, allocate space on the disk, and initialize the file. When you enter the file type, you create a control file on disk that corresponds to the data file and describes its structure and contents. The initialization procedure establishes each data file by initializing the areas on disk where data will be stored. Once a file is initialized, it can be used in other IDEAS system modules.

FN 01 -- REVISE

The Revision utility enables you to redefine a file that was defined using the Creation utilities. These procedures display all screens seen in the Creation utilities with the previously entered information. At each screen, you can accept the information as it appears, or edit the screen by stepping through each entry and changing those that are incorrect.

If you alter an initialized file using this utility, you must reinitialize the primary and the alternate files. Failure to do this results in improper control information and probable loss of data when subsequently entered. Following the reinitialization procedures destroys all data currently stored in a file. Therefore, you should use this utility only prior to data entry. If you wish to keep the data in a file that is being revised, refer to instructions in Section 3.4.

FN 02 -- DOCUMENT

The data file Documentation utilities provide necessary information about the contents and structure of data files that you may find useful when creating applications that use the data file. The documentation produced describes the file, the number of fields, the key field, the number of records specified, and the number of records on file. The documentation includes a listing of fields in the data record in both alphabetical and positional order.

FN 03 -- INITIALIZE

Touching FN 03 allows you to proceed directly to the initialization screen of the data file Creation utilities.

FN 04 -- DOCUMENT (BATCH)

After touching FN 04, you can indicate that you wish to document a series of data files. You can select any combination of the Version, Application, and Function fields to specify which data files to document. You can document only those files to which you have edit/document privileges.

FN 05 -- COPY A DATA FILE CONTROL FILE

If you wish to define a new data file that is identical or similar to an existing file, you can use this option to make a copy of the control file for the data file. You can then use FN 01 to revise this file to your specifications.

3.2 DATA FILE CREATION

Touching FN 00 from the Data File Utility screen evokes the Naming New Data File screen, Figure 3-2.

```
IDEAS Data File Editor - Access Security Control Module          Release 2.1


          Please enter the file name for the file to be created     ■■■■■■■■
```

Figure 3-2. Naming New Data File

The screen requests the file name of the file being created. You must enter the file name in the eight blocks under which the cursor is located. The file name can be from one to eight characters long. If the file name is less than eight characters, you must enter a RETURN when the name is complete in order to complete this entry; if the name is a full eight characters, entry is automatically terminated after the eighth character. If the file name already exists, an error message is displayed; you can give a different file name (EDIT) or abort the current procedure (FN 31). If you abort the procedure, the system returns to the Data File Utilities menu. Refer to Section 3.4 for instructions on modifying an existing data file.

When you name a data file, at least one of the characters must be an uppercase letter. This is necessary because the system creates two files, the actual date file and a companion control file that contains information about the file. The companion file has the same name as the data file, except that all letters in the name are lowercase. The uppercase letter in the file created by the user serves to distinguish the data file from its control file, created by the system.

NOTE

Avoid IDS2- or ids2- as the first four characters of a file name, since most system files incorporate this string in their own names. Using these characters may also result in data file address conflicts.

3.2.1 Data File Creation Screens

After you specify a file name, the Data File Creation screen, Figure 3-3, appears.

```
IDEAS Data File Editor - Initial File Specification Module          Release 2.1

File name    MAILLIST  File type ■  Disk /          Revision number        by
Description                                         Last revision date     /  /

Version              Edit/Document Privilege        Data file access privilege
Application          User class / ID      /         Read  user class/ID    /
Function             Password                       Write user class/ID    /

Primary file name                        Description

   List of valid file types                Duplicate  Random   Sequential  Alternate
                                             keys     access    access       keys
   File type    Data location               allowed   allowed   allowed     allowed
1 Reference    None, keys only                No        Yes       Yes         No
2 Primary      Data segments in file          No        Yes       Yes         Yes
3 Primary      Data segments in file          Yes       Yes       Yes         Yes
4 Primary      Key index element              No        Yes       Yes         No
5 Alternate    Associated primary file        No        Yes       Yes         No
6 Alternate    Associated primary file        Yes       Yes       Yes         No
7 Alternate    Associated primary file        Yes       No        Yes         No
```

Figure 3-3. Data File Creation Screen

First you must enter the file type of the file you are creating. The file type parameter is a single digit keyed to an accompanying table that determines whether the file is a primary, alternate, or reference file, and whether duplicate keys, random access, sequential access, and alternate keys are allowed. The seven allowable file types are indicated on the screen. Uses of these file types include:

- Type 1 -- A file containing only key fields; it cannot contain any other fields. A Type 1 file can only consist of up to five fields and must have a total length no greater than 80 consecutive positions. This file can be used to maintain a list, or to provide a table for a field validation. (Refer to Section 5.3.)

- Type 2 -- A primary data file allowing no duplicate keys, such as a customer file keyed by social security number. This is the most common data file.

- Type 3 -- A primary data file allowing duplicate key fields, such as a customer filed keyed by last name.

- Type 4 -- A primary file in which the entire record is small enough to be contained in the key index element or stored next to the key on disk. A Type 4 file can be used if the packed key length, the protect byte (1 byte), and the packed record length (including repetition of the key) together do not exceed 83 bytes. When this is the case, establishing the file as a Type 4 file saves one disk read. No alternate keys can be defined for a Type 4 file.

- Type 5 -- An alternate key file allowing no duplicate key fields.

- Type 6 -- An alternate key file allowing duplicate key fields.

- Type 7 -- Similar to a Type 6 file, except that a Type 7 file permits only sequential processing. If only sequential processing is performed on an alternate key file, defining the file as a Type 7 file saves execution time.

  An alternate file with many duplicate keys is suitable for a Type 7 file, for example a mailing list keyed by zip code. Defining the file as a Type 7 file saves time in inserting or deleting information and avoids collisions in storing the data.

You can create an alternate key file only after its corresponding primary file is fully defined.

After you enter the file type, the cursor moves to the next field. This field indicates the disk address for storing the file. The system defaults to the device address entered as Device 07 on the Peripheral Device Address Selection screen (Figure 2-6), which is the first application data file device address. You can override this default address with any address specified as Devices 07 through 15 on the Device Address Selection screen.

When you accept the device address, the system fills in the box on the upper right corner of the screen that indicates when the file was defined or last revised as well as which user ID was the last to access the file. This box is updated automatically with each revision.

The remaining fields on the screen, in the order of field entry, are as follows:

DESCRIPTION
You can enter any convenient description of the file. This description appears only in documentation and has no effect on the structure of the file.

VERSION, APPLICATION, FUNCTION
You can use these three fields to set up a filing scheme for the application (refer to Section 2.2 for more detail). All components of each application should have the same Version, Application, and Function fields if you use the IDEAS application installation utility. Section 11.10 describes this utility. Version is a one-character field, Application is a three-character field, and Function is a five-character field. You can bypass these three fields.

EDIT/DOCUMENT PRIVILEGE
You can enter three security checks to restrict who may revise or document the data file. First, you can specify a minimum user class of 0 through 9 or A through F. Second, you can specify that only a particular user can edit the file. Finally, you can require that a password be given to edit the file. You can bypass one or all of these three fields, defaulting the minimum user class to 0.

CAUTION

If you assign a higher user class than your own, or assign another's user ID for the edit/document privilege, you will be able to continue to specify the current file during the current operation. Subsequently, however, you will be unable to revise or document the data file. The same caution applies to assigning a password and then forgetting it.

DATA FILE ACCESS PRIVILEGE
These fields specify the minimum user class or particular user ID that can read from the data file or write to the data file. You can bypass these fields, defaulting the user class to 0. For alternate key files, these fields default to the values of the primary key file.

PRIMARY FILE NAME, DESCRIPTION
When you are defining a primary key file, these fields default to the name and description entered at the top of the screen. When you are defining an alternate key file, you must enter the associated primary file in this field. The system displays the description field from the description entered during primary file definition.

After you accept the entire screen for a primary key file or a reference file, the Data File Field Definition screen, Figure 3-4, appears. After you accept an alternate key file, the system proceeds directly to the Key Field Selection screen, Figure 3-5.

```
IDEAS Data File Editor - Field Attribute Edit Module            Release 2.1
File    MAILLIST                               File MAILLIST field names Pg 1 of 0
        CUSTOMERS AND ADDRESSES
FN Attribute
01 Field name
02 Sub field of
03 If sub-field, starts at byte
04 Length
05 Position in record
06 Valid character type
   0=Digits             4=U/C letters
   1=Digits & signs     5=U/C & digits
   2=Digits & dec.      6=U/C, num, & punct
   3=Any numeric        7=Any character
07 Justify
08 Zero fill
09 Decimals  ( 0-7 )
10 Security user class ( 0-9 , A-F )
FN Action               FN Action
00 Add field            19 Show name page 3
EDIT  Edit Field        20 Re-arrange for
17 Show name page 1        optimum packing
18 Show name page 2     25 Delete field
Touch desired FN key in LOWER box, FN'31 = CANCEL
```

Figure 3-4. Data File Field Definition Screen

The procedure for defining the fields in a data file is as follows:

1. Touch FN 00, as indicated in the box in the lower left corner of the screen.

2. The cursor moves to the Field name field, associated with FN 01, and the system supplies the default field name of FIELDXXX, along with other default attributes for the field. After you accept the default field name or enter a new field name, the cursor moves to the Length field, associated with FN 04. You must enter a length since a field cannot be of length zero, which is the default field length.

3. A message at the bottom of the screen allows you to accept the file as it is with the default field attributes as shown. To change any of the field attributes, touch the function key associated with that attribute. A description of each of the definable field attributes follows Step 4.

4. Accept the field by touching EXECUTE, and it appears in a list of defined fields on the right half of the screen.

Field option definitions follow. Note that for any field with only two possible conditions, touching the appropriate function key changes the condition.

FN 01 -- FIELD NAME
This is the name of the field within the data file being defined currently. The name defaults to FIELDXXX. You should override this default with a more mnemonic field name.

FN 02 -- SUBFIELD OF
IDEAS Release 2 allows you to divide a field into subfields so that you can enter, validate, sort, and process fields on the basis of their subfields. For example, the field NAME can have a subfield LASTNAME. If the field you are defining is a subfield, Touch FN 02 after entering its length and enter the field it is part of. This entry must be a previously defined field. You can nest subfields to any number of levels.

FN 03 -- IF SUBFIELD, STARTS AT BYTE
Enter the relative position in the field at which the subfield starts. For example, if the field is a phone number 1112223333, the subfield area code begins at byte 1, the next subfield of three digits begins at byte 4, and the subfield of four digits begins at Byte 7.

FN 04 -- LENGTH (REQUIRED ENTRY)
This is the length of the field.

FN 05 -- POSITION IN RECORD
This is the byte position in the data record at which the currently defined field begins. This defaults to a sequential ordering of fields based on the order in which the fields were entered. You can override the default and position the fields in the record in whatever order you wish. If you return to edit a field and increase its length, its position in the record will overlap with the next field, which is not adjusted automatically. When this occurs, you must either touch FN 20, "Rearrange for optimum packing," to rearrange fields, or adjust all following fields manually. Using this option, you can intentionally leave a gap between fields (for example, to allow room for expansion of zip code fields).

FN 06 -- VALID CHARACTER TYPE

The character set from which the characters are accepted as legal entries for this field. The different field types are:

Input	Subset Type	Input Characters Allowed
0	Digits	0 through 9, (Numeric)
1	Digits and signs	0 through 9, + (Plus), - (Minus)
2	Digits and dec.	0 through 9, . (Decimal point)
3	Any Numeric	0 through 9, +, -, .
4	U/C Letters	Uppercase (A through Z)
5	U/C and Digits	Uppercase (A through Z), 0 through 9
6	U/C, Num, & Punct	As in number 5 and punctuation characters: HEX(20) to HEX(5F)
7	Any character	As in number 6 and lower case letters: HEX(20) to HEX(7A)

Validation of characters is performed at keystroke time; the system will not allow the operator to enter an illegal character at runtime. If a field is defined as uppercase only, the system automatically translates lowercase to upper case characters at data entry.

FN 07 -- JUSTIFY

The field can be right-justified within the record. A Y indicates right-justify, an N indicates left-justify only. You can specify this for alphanumeric fields as well as numeric.

If you are using a numeric field as a key on which to sort a report, you must specify that the field be right-justified. If you do not, the number 100 will print before the number 2.

FN 08 -- ZERO FILL

You can pad a field that is right justified with zeros on the left. If you want a numeric field to print on a screen with leading zeros (for example, for zip code 07748 to print as 07748 instead of 7748), you must specify the field as zero-filled.

You should specify only numeric fields (Types 0, 1, 2, and 3) as zero-filled.

FN 09 -- DECIMALS (0 through 7)

You can specify a number of decimal places for a numeric field. At runtime, the number entered by the operator is rounded to the specified number of places if it contains a decimal. If the number does not contain a decimal point, the implied decimal point is assumed to be N places to the left of the last digit entered, where N is the number of specified decimal places.

If you specify 0 decimal places (the default value), then the number entered is not adjusted, but is saved in the form in which it is entered.

FN 10 -- SECURITY USER CLASS

All users of an IDEAS Release 2 system belong to one of 16 hierarchical user classes (from 0 through 9 and A through F, with F being the highest and 0 being the lowest). This entry designates the minimum user class that can access this field. For example, a user of Class 5 can access any field designated as accessible to fields 0, 1, 2, 3, 4, or 5, but cannot access a field designated as user class 6 through 9 or A through F.

In addition to adding new fields to the data file, you can perform the following actions, as indicated in the box in the lower left corner of the screen:

EDIT -- EDIT FIELD

You can modify a field that has already been defined by touching the EDIT key, entering the name of the field to be modified, and proceeding as if defining a new field.

FN 17 THROUGH FN 19 -- SHOW NAME PAGE 1, 2, AND 3

IDEAS allows for up to 249 fields per data file, but if more than 84 fields are defined, they cannot fit on one screen. Instead, the system displays the first 84 fields as page 1, the second 84 fields as page 2, and the final 81 fields as page 3. To change the page being displayed, touch FN 17, FN 18, or FN 19.

FN 20 -- REARRANGE FOR OPTIMUM PACKING

This option sorts the fields in the data file so that numeric fields are first, uppercase fields second, and all other fields third. Numeric fields are packed 2:1 and uppercase fields are packed 4:3. This changes the positions of the fields within the record. You should evoke this option unless there is a specific reason for another field order.

FN 25 -- DELETE FIELD

To delete a defined field, touch FN 25, enter the name of the field to be deleted, then touch FN 09, as indicated in the prompt at the bottom of the screen. Deletion of any field other than the last (physical) field in the record or a subfield will result in a "hole" in the record where the field had been located. For example:

Field	Len	Beg	End
ACCOUNT#	6	1	6
BALANCE$	10	7	16
ADDRESS	30	17	46
NAME	30	47	76

Delete "ADDRESS"

ACCOUNT#	6	1	6
BALANCE$	10	7	16
NAME	30	47	76

Positions 17 through 46, where ADDRESS was, are no longer used, but space for them is still left in the record. Because of this, you can control the placement of fields within the record. You can save this space manually by repositioning subsequent fields or by using the "Rearrange for optimum packing" routine.

After you define all fields and subfields in the date file, accept the screen and the Key Field Selection screen, Figure 3-5, appears.

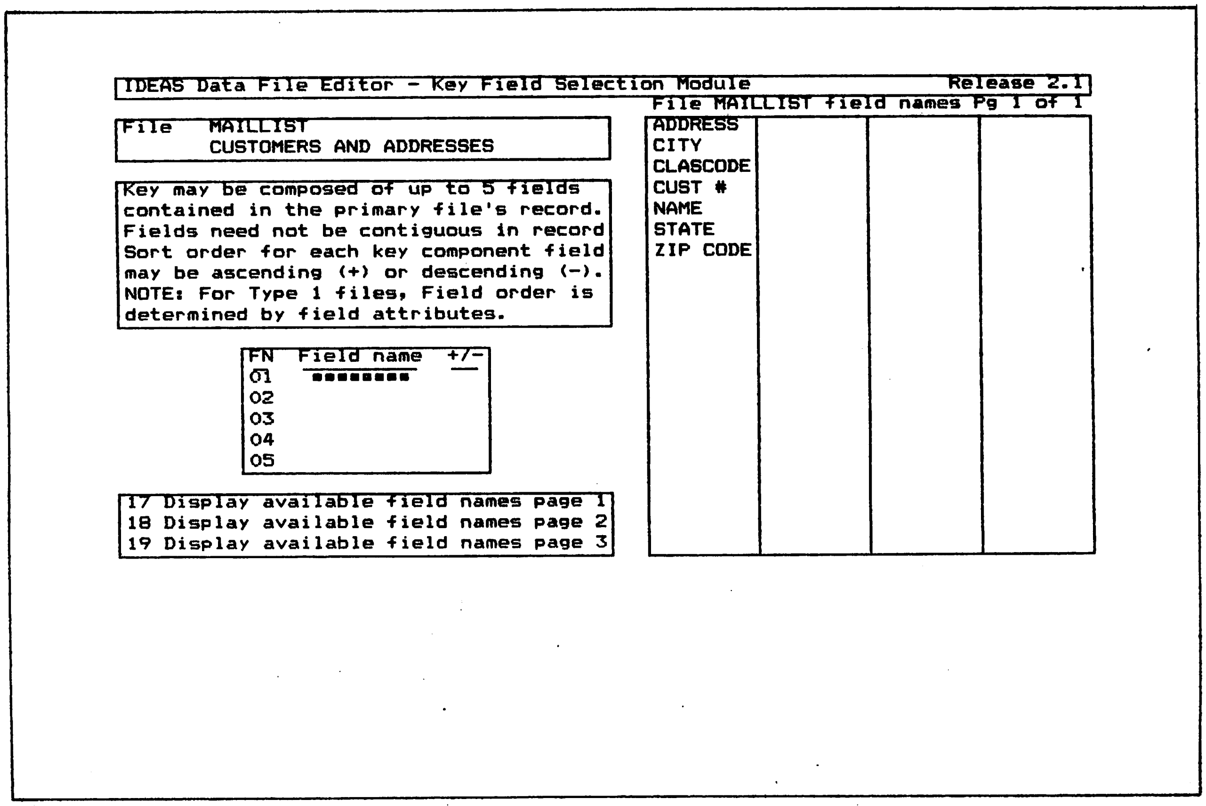


Figure 3-5. Key Field Selection Screen

Up to five fields within the data record compose the key for a data file. These fields need not be contiguous, but can total no more than 80 bytes in length. The system displays page 1 of the field names for the data file under consideration on the right half of the screen; touching FN 18 or 19 displays additional fields.

After entering each key field, indicate whether you will use the field in ascending or descending sort order. A + character following the field name, the default value supplied by the system, indicates ascending order; a - character indicates descending order. The sort order characters determine the order in which records return during a sequential scan of the file. If, in different circumstances, you wish the same records to return in either ascending or descending order, you must create an alternate file with the same key but opposite sort order.

For a Type 1 file, you must define the key fields in the order in which they appear in the record. The first field in the record must be the first key, the second field in the record must be the second key, and so on. The key for a Type 1 file must consist of all fields in the file.

Touch the indicated function key to add additional key fields or to change a key field already entered. After you accept the key fields, the Disk Space Allocation screen, Figure 3-6, appears.

```
IDEAS Data File Editor - Disk Space Allocation Module           Release 2.1

File  MAILLIST    CUSTOMERS AND ADDRESSES           'FN  Operation
                                                    '00  Change # of volumes*
File Type 2                 Number of volumes 1     '01  Edit volume # 1**
(Note: Only file types 2 & 3 may be multi-volume.)  '02  Edit volume # 2**
                                                    '03  Edit volume # 3**
Records Specified ********   No. provided           '04  Edit volume # 4**
                                                    '05  Edit volume # 5**
                  Sectors     Minimum    Sectors    '06  Edit volume # 6**
Vol    Disk      Available    Required   Allocated  '07  Edit volume # 7**
 1      /                                           '08  Edit volume # 8**
 2      /                                           '15  Change # of records
 3      /                                           '16  Change vol 1 address
 4      /                                           '31  Cancel
 5      /                                           * File types 2 & 3 only
 6      /                                           **Applicable only to the
 7      /                                             no. of volumes selected
 8      /                                             & only to multi-volume
 Total sectors                                      Required must = Allocated
```

Figure 3-6. Disk Space Allocation Screen

All of the information you have entered until now establishes a data file control file. The data file itself is not created until you accept the Disk Space Allocation screen.

The first parameter the screen requests is the number of records you will place in the file. The number supplied should be the maximum number you expect the file to contain. The system displays the number of records it actually provides, usually 5 percent more than you requested. This extra allocation minimizes overflow conditions when the file reaches stated capacity. Alternate key files should always request the same number of records as the primary key file, to ensure compatibility.

After displaying the number of records provided, the system displays the number of sectors available on the disk indicated on the Data File Creation screen (Figure 3-3). The system also displays how many sectors the data file requires, and how many sectors the system has allocated. A single volume of a data file cannot be larger than the space available on a single disk. If a conflict arises, the system displays an error message and you must change the appropriate parameters.

You can change the total number of volumes by touching FN 00. When more than one volume is indicated, all keys lie in the first volume. For this reason, alternate key files (File Types 5, 6, and 7) cannot span multiple volumes, and files containing only keys (Types 1 and 4) cannot span multiple volumes. This is also the reason for the minimum requirement for Volume 1.

After you indicate how many volumes you need, the system displays how many sectors of each additional volume it requires for the number of records specified. Touch the appropriate function key for each additional volume, enter the device address for that volume (which must be an address specified on the Peripheral Device Selection screen, Figure 2-6), and enter how many sectors in that volume to allocate to the data file. A minimum number of sectors in each additional volume, indicated on the screen, must be devoted to the data file. The number of sectors allocated in the additional volumes must be this minimum number or a multiple of this number.

Note that the total number of sectors allocated must equal the total number of sectors required. It will probably take some manipulation of the number of records required and the number of sectors allocated per disk before the two numbers are equal. IDEAS sounds an audio alarm each time you attempt to accept an allocation screen without allocating all space properly.

After you enter the number of sectors in the final volume, you may accept the screen or edit any of the parameters on the screen by following the instructions displayed on the screen. Touching RETURN accepts the screen as shown and proceeds to the Data File Initialization module.

3.2.2 Data File Initialization Screen

The final step in creation of a data file is initialization. Initialization is the process of setting aside the disk space specified for a file and creating empty record positions using the information in the data definition file as a guide. A single screen is involved in this module, but you can initialize several files (i.e., a primary data file and its associated alternate key files) at a time. When the screen first appears, the default file name is the one that was specified when you first entered the creation module. Any associated files (alternate key files for the primary file) which have been defined are displayed also and can be initialized along with the default file. Any of the associated files whose names appear on the screen also can be initialized singly.

It is possible to put this screen on an application menu, allowing the user of an IDEAS-application to initialize data files at runtime. Refer to Section F.4 for a list of the IDEAS files necessary to do this.

```
IDEAS Data File Editor - File Initialization Module          Release 2.1

          FN  File type    Name      Description
          00  Primary   2  MAILLIST  CUSTOMERS AND ADDRESSES
          01  Alternate
          02  Alternate
          03  Alternate
          04  Alternate
          05  Alternate
          06  Alternate
          07  Alternate
          08  Alternate
          09  Alternate
          10  Alternate
          11  Alternate
          12  Alternate
          13  Alternate
          14  Alternate
          15  Alternate
          16  Alternate
          17  Initialize all files listed above, primary & alternates

          Now initializing file          Processing is    % complete
Touch EXECUTE to initialize MAILLIST, or CANCEL to abort
```

Figure 3-7. Data File Creation - File Initialization Screen

You should initialize all related files (primary data file and associated alternate key files) simultaneously. This saves time and ensures that an alternate key file is not inadvertently left uninitialized. Therefore, you should define all alternate key files before initialization occurs. You cannot define alternate key files before the associated primary file has been defined.

CAUTION

> Initializing any IDEAS data file causes the destruction of all information contained in that file. For instructions on retaining information in a file that is being reinitialized, refer to Section 3.4.

To abort the current operation, touch FN 31. This returns control to the Data File Utilities menu. Touching EXECUTE at this time will cause only the default file to be initialized. Touching FN 00 through FN 16 initializes individual associated files, and FN 17 initializes all files.

After you have accepted or canceled the file initialization screen, the system returns to the Data File Utilities screen.

3.3 DATA FILE DOCUMENTATION

After a data file is initialized, you can generate documentation for that file. Documentation of each file is important in construction of related files as well as screen masks and application programs, since information provided during documentation is needed during entry of other parameters.

Enter the Data File Documentation module by touching FN 02 from the Data File Utilities menu. You must then supply the name of the file to be documented in the blocks displayed. If, however, you have just finished creating or altering a file, that file name is displayed as a default. Touching RETURN will cause the system to use the default name supplied; touching any other printing key will clear the default name and allow you to supply the desired file name.

The system loads the Data File Documentation screen (Figure 3-8).

```
IDEAS Data File Editor - File Documentation Module                Release 2.1
File MAILLIST T=2 Rev#     0 092783 bySJL    File MAILLIST field names Pg 1 of 1
Desc CUSTOMERS AND ADDRESSES                 ADDRESS
Version 1 Application AAA Function DEMO      CITY
                                             CLASCODE
Security: Read  user class 0 user ID         CUST #
          Write user class 0 user ID         NAME
E/D password          u/cl 0 user ID         STATE
                                             ZIP CODE
Records:      Specified       500    0.00%
       0      Provided        532    0.00%
  7 fields    L 0079 N 0005 A 0000 P    78

Key fld    Pos Len S   Primary      MAILLIST
CUST #       1   5 +   Alternate key files




Packed key len     3
Pointer            2
Data in KIE        0
Total KIE len      5
Touch EXECUTE to document, FN'1-FN'3 to display field name pages, or CANCEL
```

Figure 3-8. Data File Documentation Screen

The documentation screen displays the following information about the file. Appendix D presents further information about the file structure of an IDEAS file.

FILE
: The name of the file being documented.

REV# BY
: The number of times the Data File Revision utility has been entered to edit this file, the date of the last revision, and the user ID of the last person to edit the file.

DESC
: The description of the file, as entered by the developer.

VERSION, APPLICATION, FUNCTION
: Fields as entered by the developer.

SECURITY
: The user class and user ID necessary for reading from the file and writing to the file, and the password, user class, and user ID necessary for editing and documenting the file.

RECORDS
The number of records specified, the number provided by the system, and the percentage of those records that are filled with data.

FIELDS
The number of fields in the data file.

L
The total record length of the data file.

N
The number of numeric bytes that can be packed.

A
The number of uppercase bytes that can be packed.

P
The packed length of the record, which includes the protect byte that indicates the beginning of packed data.

KEY FLD, POS, LEN, S
The key fields, their byte position in the record, their length, and whether their sort order is positive or negative.

The following fields describe various components of the data file structure. For complete information on the structure of an IDEAS data file, refer to Appendix D.

PACKED KEY LEN
The packed length of the key field.

POINTER
The length of the pointer to the data.

DATA IN KIE
The length of any data in the key index element (KIE) that is not part of the key or the pointer. This is 0 for any but a Type 4 file.

TOTAL KIE LENGTH
The packed key length, plus the length of the pointer, plus any additional data.

The screen also displays the names of the alternate key files, and the fields within the record. If the field names span more than one page, you can view the extra pages by touching FN 02 and FN 03.

Touching EXECUTE prints the data file documentation at the currently selected printer. Touching FN 31 returns the system to the Data File Utilities menu.

3.4 DATA FILE REVISION

After a data file is defined, you can revise it using the Data File Revision utility. To access this utility, touch FN 01 from the Data File Utilities menu (Figure 3-1). In general, the module proceeds from screen to screen exactly as if you were creating the file. The system displays each screen with the previously entered file information. You can then accept each screen as displayed, or edit the screen.

If you wish to revise an initialized file, you must reinitialize when revision is complete. In the case of a primary data file, you also must reinitialize all associated alternate key files. If you wish to revise an uninitialized file, you can treat it as if you were creating the file.

When you enter the module, enter the name of the file to be revised. The name is not subject to revision. After you enter it, you can begin to revise the file's parameters.

After you touch FN 01 from the Data File Utility screen, the Name Specification screen (Figure 3-2) appears. Enter the name of the data file, and the Data File Creation screen (Figure 3-3) appears. If a password is associated with this data file, you must enter the password at this point.

When the Data File Creation Screen appears, the prompt "Touch EXECUTE to accept and save Control File, EDIT to modify, FN 31 to CANCEL" appears. You can bypass any alterations to this screen and proceed to the next by touching EXECUTE. If you wish to change any parameters here, however, touch EDIT.

When you revise a primary file, the Data File Field Definition screen (Figure 3-4) follows address/type revision. Again, you must accept or revise the currently displayed screen. When you revise an alternate key file, you bypass this screen because the field definition occurs in the primary file.

If you change the length of a field in the data file when you revise the file, subsequent field positions are not updated unless you use the "Re-arrange for optimum packing" option of the Field Definition screen or reposition each subsequent field in the file manually. This could cause overlap of field positions if not taken into account.

Revising a data file after defining the screen and report programs that use that data file could alter those programs. You should return to the screen and report files after redefining a data file and make sure that all field attributes are correct, or run the Batch Screen Update utility, described in Subsection 4.2.3.

Once you edit or accept the data record (or bypass the screen, in the case of alternate key files), you can respecify the key fields. The Key Field Selection screen (Figure 3-5) appears, with keys and sort orders specified and the accept/edit prompt displayed.

> **NOTE**
>
> If you alter any field location or size in the primary data file, you must perform the Key Field Revision section of this module for both the primary file and <u>all</u> alternate key files. To do this, accept the Key Field Selection screen. Failure to do this will result in incorrect information pertaining to the location of fields within the record and serious errors later on.

<u>Whenever</u> you alter a file, <u>you must reinitialize it</u>. If you are changing an alternate key file, you need only reinitialize that file, but if you are changing a primary file, you must reinitialize all associated alternate key files as well. For instructions on initializing files, refer to Section 3.2.

When a file is reinitialized, all data contained in that file is lost. If you wish to keep the data in a file that is being revised, use the Batch Program option of the IDEAS Primary Program Selection screen to create a batch program. This program should read from the original file and write to a work file in a revised format. After reinitialization, you can write a batch program to write the data back to the original file.

If you create a new alternate key file or reinitialize an old one when there is data in the file, you should fill the new file with appropriate data (pointers to the data in the primary file). To do this, choose the Key File Reconstruct utility from the Other Utilities option on the Primary Program Selection menu.

3.5 BATCH DOCUMENTATION

When you touch FN 04 from the Data File Utilities screen, The Data File Batch Documentation screen (Figure 3-9) appears.

```
IDEAS Data File Editor - Data File Batch Documentation Module        Release 2.1

        Note:  This module will document all IDEAS data
               files which meet the following criteria:

            1. A password is not required for documentation
            2. The edit/document user class is <= yours
            3. The user ID, if any, is yours
            4. The version, application code, and function code
               are as specified below.  A blank entry in any of
               these 3 will cause that parameter to be ignored.

                   Version           =

                   Application code  =

                   Function code     =
```

Figure 3-9. Data File Batch Documentation Screen

You can use this option to document a series of data files, as determined by the user-defined parameters Version, Application, and Function. The screen specifies what other criteria are necessary for inclusion in batch documentation.

3.6 COPYING A DATA FILE CONTROL FILE

If you wish to define a new data file that is identical or similar to an existing file, you can use this option to make a copy of the control file for the data file. You can then use FN 01 to revise this file to your specifications.

When you touch FN 05 from the Data File Utilities screen, a screen appears asking the name of the data file whose control file you wish to copy. After you enter the name of this data file, a message appears asking for the name of the copy of this data file. Enter a name, press execute, and the system makes a copy of the control file under the new name and returns to the Data File Utilities screen. After this occurs, you can revise, document, or initialize the new data file.

NOTE

If a data file control file specifies an incorrect data file address (a situation that would occur if you move a data file and its control file but do not modify the control file), a message appears at runtime indicating this, and you can enter the correct address. The control file, however, must be at the address indicated in the START module for the application.

CHAPTER 4
SCREEN UTILITIES

The Screen utilities constitute the major component of IDEAS Release 2 because of the two important functions they perform. First, the Screen utilities enable you to define formatted screen displays (masks) through which you can implement data entry and inquiry. Second, the Screen utilities define the sequence of field processing performed on the data fields, both before and after data entry. The screen field edits and their associated options form what amounts to a programming language. Through this "language," you determine the flow of control for the data processing and indicate what conditional actions to perform under what circumstances. The variety of activities IDEAS supports is extensive, and enables an IDEAS user to develop an elaborate application without writing any BASIC code.

The instructions for using the Screen utilities are divided between Chapters 4 and 5. Chapter 4 is divided into two sections. The first section contains a screen-by-screen explanation of the method of defining a screen mask and the fields on the screen. It includes a section on how to use the Screen Mask editor. Information about help screens, subscreens, screen buffers, and use of function keys in the defined screen is given in context. The second section describes the additional Screen utilities and includes information on archiving screens and retrieving screens from archive. Chapter 5 describes the capabilities and features of field processing, discussing the Edit Specifications and how to implement programming logic.

When you choose the Screens option from the Primary Program Selection screen, the Screen Mask Utility menu, Figure 4-1, appears.

```
IDEAS Screen Mask Editor - Utility Selection Menu Module          Release 2.1

                              FN Program Module
                            ■ 00 Create
                            • 01 Revise
                            • 02 Document
                            • 03 Extra screen utilities
                            • 04 Document (batch)
                                 -- Other Functions --
                            • 06 Data files
                            • 08 Programs (interactive)
                            • 09 Programs (batch)
                            • 11 Reports
                            • 12 Menus
                            • 13 START programs
                            • 14 Other utilities
                            • 15 Applications
                            • 31 Main IDEAS menu

SEP 27 83                                                        Term #  2
Mem 36.00                      Global Selected                   Sta. # 20

Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 4-1. Screen Mask Utility Menu

Summaries of the first five options on the Screen Mask Utility menu follow. Sections 4.1 and 4.2 explain FN 00 in detail. Section 4.3 describes the remaining options.

FN 00 -- CREATE

The screen mask creation module allows you to define a screen by typing the text that is to appear in the display at the desired location in the screen. You must also indicate where data entry fields appear on the screen and define the attributes of those fields. You define all field processing with this utility.

FN 01 -- REVISE

The screen mask revision module permits you to recall and modify an existing screen mask and the fields associated with that screen.

FN 02 -- DOCUMENT

The screen mask documentation module permits you to print a copy of a screen mask and a listing of field attributes for each data field in the display. Documentation also describes all field processing associated with the screen.

FN 03 -- EXTRA SCREEN UTILITIES

The extra screen utilities enable you to archive screens along with their associated edits and to retrieve the screens from the archive. When archiving a screen, you have the option of deleting screen edits from the Edit Specification file, allowing room for additional edits in the file. The utilities also allow you to copy a screen under a different name, thereby allowing you to reproduce a screen without reentering it. A Batch Screen Update utility allows you to update the data file field specifications in a series of screens.

FN 04 -- DOCUMENT (BATCH)

The Batch utility enables you to document a series of screen files. You can indicate any combination of Version, Application, and Function to specify which screens to document, but you can document only those files to which you have security access.

4.1 DEFINING A SCREEN

Touching FN 00 loads the Naming Screen Mask screen, Figure 4-2.

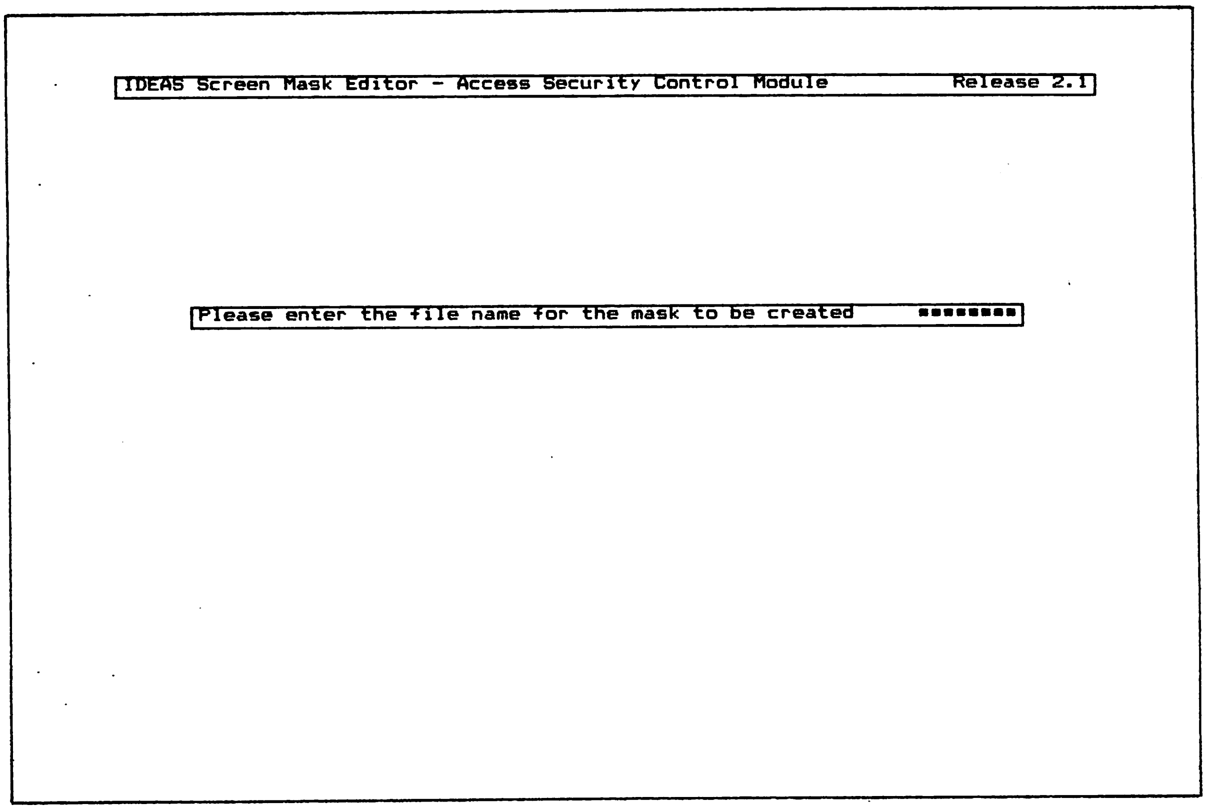


Figure 4-2. Naming New Screen Mask

The prompt shown in Figure 4-2 requires the name of the screen being created, which may be from one to eight characters in length. The Screen Mask Specification screen, shown in Figure 4-3, appears after you enter the file name.

```
IDEAS Screen Mask Editor - Screen Mask Specification Module            Release 2.1

Screen name MAILSCRN                             Last revision no.          by SJL
Description ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■     Date of last revision 09/27/83

Version              Edit & document privilege   HELP screen (if any)
Application          User class     User ID      FN keys to be trapped
Function             Password                    (up to 8 may be used)

      Data files associated with screen             General screen attributes
FN    File    Description                        FN Attribute
01                                               08 This is a sub screen       N
02                                               09 Skip ahead keys enabled    Y
03                                               10 Skip back keys enabled     Y
04                                               11 Cancel key enabled         Y
05                                               12 Clear work buffer @ load   Y
06                                               13 Display all fields @ load  N
07                                               14 First line is bright       Y

Minimum size of work buffer is    1750 bytes     Number of fields on screen    0
Minimum size of record buffer is   256 bytes     Number of boxes on screen     0
```

Figure 4-3. Screen Mask Specification Screen

4.1.1 Screen Mask Specifications

The fields on the screen, in the order of field entry, are:

DESCRIPTION

You can enter any convenient description of the screen. This description must be entirely uppercase.

VERSION, APPLICATION, FUNCTION

Refer to Section 2.2.

EDIT & DOCUMENT PRIVILEGES

You can enter three security checks to restrict who may revise or document the data file. First, you can specify a minimum user class of 0 through 9 or A through F. Second, you can specify that only a particular user can edit the file. Finally, you can require that a password be given to edit the file. You can bypass one or all of these three fields, defaulting the minimum user class to 0. Note the caution in Section 3.2 about assigning a higher user class than your own.

HELP SCREEN

If you enter a screen name in this field, that screen becomes a help screen associated with the screen currently being designed. An operator can display help screens at runtime by touching FN 15. After display of the help screen, control returns to the original screen with the cursor in the same position and all data intact. To qualify as a help screen, a screen must have no keyboard-modifiable fields specified.

A help screen is not the same thing as a subscreen, which is defined below. Help screens are defined separately from their associated screen using the Create option of the Screen Mask Utility menu.

FN KEYS TO BE TRAPPED

In an IDEAS-developed screen, data field type or screen field type determines valid entries to that screen at runtime. You can enable the skip-ahead and skip-back keys and the CANCEL key (FN 31) as valid entries for a particular screen, as described below. The help screen key (FN 15) is also a valid entry. In addition to these, you can specify up to eight other keys as FN keys to be trapped, enabling them as valid entries for the screen.

You must specify function keys as character pairs: 01, 02, 03, 04, through 31. You can also trap the EDIT key by specifying 33, the FN or TAB key by specifying 34, and the shifted FN or shifted TAB key by specifying 35. On a 2236DW or 2336DW terminal, IDEAS translates the INSERT key to FN 10 and the DELETE key to FN 09. RUN and RETURN are not considered function keys, and thus cannot be trapped.

NOTE

When you enter a trapped function key at runtime, processing branches to pre-entry Edit Specifications on the field on the screen with the last field number. Field numbers, which determine sequence of field processing, are described along with FN 01 in Subsection 4.1.3. For general information on Edit Specifications and use of trapped function keys, refer to Sections 5.1 and 5.2.

You also can use this option to override any preset use of an enabled function key, because setting function key traps takes precedence over pre-set use. For example, if the CANCEL key is enabled on the screen but you specify FN 31 as a function key to be trapped, touching FN 31 does not automatically cancel out of the screen.

Function keys can be trapped at the field level also (refer to the discussion of Option Y in Section 5.5). A field-level function key trap has execution priority over screen-level function traps.

The cursor does not go to the following fields automatically. You can change the default values for these fields by touching the indicated function key.

DATA FILES ASSOCIATED WITH SCREEN

A screen can access up to seven data files. To indicate an associated data file, touch FN 01 through 07 and enter the name of the data file. An area equivalent to one record from that data file is set aside in the screen work buffer to be used when you read a record. When you first create a screen mask, the key field(s) from the file you establish as File 1 appear automatically.

The buffer size fields at the bottom of the screen are information-only fields that tell the screen designer the minimum required buffer sizes to handle all of the associated data files. You should note this information and use it to establish the work buffer size when running the START program creation module.

If you need to retrieve a record through the primary or alternate key, you must specify the primary and alternate key files separately.

You need not list in this area files from which you want individual fields for reference only. You can obtain these fields using the Read Record Operation of the Edit Specifications. Refer to Section 5.3 for information on this option.

FN 08 -- THIS IS A SUBSCREEN

A Y response indicates that the screen is to be used as a subscreen entered from another screen. An N response to this parameter clears the previous CRT screen and displays all screen text prior to processing of the first defined field. If you specify a screen as a subscreen, all text from the previous screen image remains displayed on the CRT and the subsequent screen information is overlayed. If you do not define any fields on a subscreen, the field attributes for the previous screen control user input. If you do not associate any files with a subscreen, the previous files remain to control buffer limits. When defining a subscreen, note the information given under the description of the FN/TAB key in Subsection 4.1.2 and the information given under the description of the heading "Position of Field" in Subsection 4.1.4.

FN 09 -- SKIP AHEAD KEYS ENABLED

A Y response activates skip-ahead keys prior to processing of the first field; an N response deactivates them. These keys include FN 04 END, FN 11 SKIP 5, and FN 12 NEXT FIELD. Skip-ahead keys can also be controlled during program execution with Pass/Fail Options A and B. (Refer to Section 5.5.)

FN 10 -- SKIP BACK KEYS ENABLED

A Y response activates skip-back keys prior to processing of the first field; an N response deactivates them. These keys include FN 07 BEGIN, FN 13 LAST FIELD, and FN 14 BACK 5. Skip-back keys can also be controlled during program execution with Pass/Fail Options C and D. (Refer to Section 5.5.)

FN 11 -- CANCEL ENABLED

A Y response enables the CANCEL key (FN 31) during data entry; an N response disables it. If you enable the CANCEL key, you can leave the data entry screen at runtime and return to the menu that called it by touching FN 31. You can also enable or disable the CANCEL key before and after processing individual fields with Pass/Fail Options E and F, and you can change the module to be loaded when you touch CANCEL key using Pass/Fail Option S. (Refer to Section 5.5.)

If you enable the CANCEL key press FN 31, any protected records become unprotected. Record protection is described in detail in Section 5.3.

FN 12 -- CLEAR WORK BUFFER AT LOAD

A Y response indicates that the screen work buffer clears prior to processing of the first field. This means the area used to store input to screen fields is set to blanks. If the screen is the first screen you encounter after a menu, set this response to Y to clear information left in this buffer from the menu. You can use a response of N, for example, for a subscreen that shares fields with the main screen.

FN 13 -- DISPLAY ALL FIELDS AT LOAD

A Y response places pseudo-blanks in every character position of every displayable field prior to processing the first field. Any fields that contain information, display that information in addition to normal text and screen boxes.

FN 14 -- FIRST LINE IS BRIGHT

A Y response highlights any screen text located on Line 1. This response only affects the screen text; the intensity of fields is determined by their defined attributes. The first line appears with normal intensity if your response is N.

The following information also appears on the Screen Mask Specification screen.

LAST REVISION

no. - This number increases by one each time a screen mask is edited.
by - The ID of the last user to edit the screen mask.

DATE OF LAST REVISION

The date entered as the system date when the screen mask was edited last.

MINIMUM SIZE OF WORK BUFFER/MINIMUM SIZE OF RECORD BUFFER

The number displayed in this field indicates how large the work and record buffers must be to accommodate all of the screen fields and files. After you enter the file names and define the screen fields, IDEAS computes the minimum buffer sizes. These numbers can exceed the default values of 1750 bytes for the work buffer and 256 bytes for the record buffer: the work buffer is the sum of any data files plus any unassociated fields (with a minimum size of 1750 bytes) and the record buffer is the size of the largest record from an associated data file (with a minimum size of 246 bytes). If you require a buffer that is larger than the minimum size displayed here, note the sizes provided on the first screen of the START Program utility and increase them manually.

NUMBER OF FIELDS ON SCREEN/NUMBER OF BOXES ON SCREEN

The number of fields and the number of boxes defined for the screen.

4.1.2 Screen Mask Editor

The second screen displayed while creating or revising an application screen is the Screen Mask editor. You can use the Screen Mask editor to create the overall screen as it appears during runtime. You can place text anywhere on the screen (except locations occupied by fields and all of Row 24) simply by moving the cursor and typing text. Using FN 00, you can create variable or data entry fields. Subsection 4.1.3 describes in detail the process of defining a field. You can move text and fields around the screen and perform a variety of other control functions with the function keys, as described in the following paragraphs.

When you create a new screen that has an associated data file, the key fields for the file specified as File 1 automatically appear on the screen, along with a message. You can erase the message at this point by pressing FN 08 (ERASE). You should then move the key field(s) to the location you want with FN 21, 22, 25 and 26, as described below.

The key fields default to nonbypassable, required, bright intensity, and any attributes established during data file field definition. Refer to Subsection 4.1.3 for a description of field editing and attributes.

If you do not wish to use the key fields on your screen, delete them at this point by canceling out of the screen to the Screen Utilities menu and touching FN 01 (Revision). The name of the screen mask you canceled out of appears as the default name; touch EXECUTE, then touch EXECUTE again to accept the specification screen. This returns you to the Screen Mask editor.

The differences in the way the screen mask appears during development and the way it appears at runtime follow.

- The design screen represents spaces as single dots unless you use the FN/TAB key (see below) to redefine the dots as "null" spaces. The functional difference between dots and null spaces is important when defining a subscreen; this difference is described below under the heading FN/TAB. In both cases, the space characters are represented as blank areas on the user display.

- Text characters appear both during design and at runtime exactly as entered. Text can consist of field prompts, screen titles and/or other descriptive information to aid the user. All text appears at a normal display intensity with the exception of that located on the first line, which you can choose to display at high intensity.

- During design, screen fields appear as a series of pseudoblanks starting at the desired screen position. The number of psuedoblanks is an accurate representation of the field length. The design screen still represents fields designated as non-display with proper position and length. Numeric fields do not have any decimal representation on the screen during design.

Cursor movement using control keys (FN 04, 05, 06, 07, 11, 12, 13 and 14) does not affect text or fields already placed on the screen, and you cannot enter either text or spaces into positions already used as part of a field.

The following function keys are operational during screen mask text editing for the purposes described below. Note that on a 2236DW or 2336DW terminal, in addition to the numeric function keys, certain labeled function keys perform text editing on a screen, such as the INSERT and DELETE keys and the north, south, east, and west arrows. In these cases, using either the key of the appropriate name or the function key of the appropriate number has the same result. In the following listing, separate keys that perform the same function are separated by a slash.

FN 00 -- EDIT FIELD

Adds a field or edits a field. It evokes the Field Definition screen (Figure 4-4). Subsection 4.1.3 details the process of defining a screen.

Touching FN 00 when the cursor is under a null (blank) or a space (dot) character on the screen loads the Field editor with default values generated for a new field. (This is the procedure for adding a field.)

Touching FN 00 when the cursor is within a field loads Field editor with the existing attributes for the field displayed. (This is the procedure for editing a field.)

Touching FN 00 when the cursor is under a normal text character sounds the audio alarm.

FN 02/CENTER -- CENTER
If there are no fields on the current line, centers all text on the line; the cursor remains at the current position. Any null (blank, no dot) positions on the line are changed to spaces (dots). If there are fields on the line, the alarm sounds and the key is ignored.

FN 04 -- END
Moves the cursor to Row 23, Column 80.

FN 05/SOUTH -- DOWN
Moves the cursor down one row with the column position unchanged unless the current row is Row 23 and the current column is less than 80, in which case the alarm sounds and the key is ignored. When the cursor is at Row 23, Column 80, FN 05 moves the cursor to 24, 80, where you can define a special 1-byte field, as explained in Subsection 4.1.3.

FN 06/NORTH -- UP
Moves the cursor up one row without changing the column position unless the current row is Row 1, in which case the alarm sounds and the key is ignored.

FN 07 -- BEGIN
Moves the cursor to Row 1, Column 1.

FN 08 -- ERASE
Places spaces (dots) in all non-field character positions on the current row starting at the current column and continuing to the end of the row. It does not affect fields on the current row.

FN 09/DELETE -- DELETE
Deletes the character at the current cursor position and shifts all subsequent text on the current row one position to the left, with a space (dot) placed in Column 80. It does not affect fields on the row (refer to FN 25). If the cursor is at Column 80, the alarm sounds and the key is ignored.

You can move text using FN 09 DELETE to positions within existing fields, thus potentially overriding field text positions. If this happens, either clear screen text from field positions, using FN 08, as specified above, or return the screen text to its former location using FN 10, as discussed below.

FN 10/INSERT -- INSERT
Shifts the text starting at the current position and continuing through Column 79 on the current row one position to the right; the character formerly in Column 80 is lost. Inserts a space (dot) at the current column. Does not affect fields (refer to FN 26). You can move text using FN 10/INSERT to within existing fields, as described above under FN 09.

FN 11 -- 5 RIGHT

Moves the cursor five columns to the right unless it is in Columns 76 through 80, in which case the alarm sounds and the cursor goes to Column 80.

FN 12/EAST -- 1 RIGHT

Moves the cursor one column to the right unless it is already in Column 80, in which case the alarm sounds and the key is ignored.

FN 13/WEST -- 1 LEFT

Moves the cursor one column to the left unless it is already in Column 1, in which case the alarm sounds and the key is ignored.

FN 14 -- 5 LEFT

Moves the cursor five columns to the left unless it is in Columns 1 through 5, in which case the alarm sounds and the cursor goes to Column 1.

FN 16 -- EDIT BOX

If the cursor is at the starting position of a box (a box starts at the upper center of the current character position), FN 16 erases the box from the screen. The row, column, depth, and width of the box appear at the bottom of the screen, with the cursor in the ROW attribute field. You can now delete the box by using FN 31, or you can edit it. If two or more boxes share common lines or segments, the common portions disappear from the screen until you: enter and subsequently leave the field editor; save and enter the same screen for revision; or access the partial box for editing purposes.

If the cursor is not at the current starting position of a box, default values appear for row, column, depth, and width at the bottom of the screen as you step through these fields. Pressing FN 31 aborts the box specification.

The default attributes are:

ROW	Current cursor position
COLUMN	Current cursor position
DEPTH	Zero (specifies a horizontal line)
WIDTH	80-C, where C is the specified starting column

The term "box" in the Screen Mask editor means either a box or a horizontal or vertical line. You can specify horizontal or vertical lines by setting either the depth or the width, respectively, to zero.

FN 21 -- ROW DOWN

Moves all text and fields down one row, starting at the current column (to the right of the cursor). When the cursor is at Row 23 or there is a field or text character in part of the row below and to the right of the current column position, the alarm sounds and the key is ignored.

FN 22 -- ROW UP

Moves all text and fields up one row from the current row, starting at the current column. When the cursor is at Row 1 or there is already a field or text character in part of the row above and to the right of the current column position, the alarm sounds and the key is ignored.

FN 23/SHIFTED DELETE

Performs the same actions on text and is subject to the same restrictions as FN 09. In addition, it moves all fields on the current row to the right of the current cursor position one column to the left. If the cursor is within a field when this key is used, the alarm sounds and the key is ignored.

FN 26/SHIFTED INSERT

Performs the same actions on text and is subject to the same restrictions as FN 10. In addition, it moves all fields on the current row one column to the right, starting at the current cursor position. If the cursor is within a field or Column 80 contains a field position when this key is touched, the alarm sounds and the key is ignored.

FN 31 -- CANCEL

Reloads the Screen Mask editor menu. Changes made to the Edit Specifications are saved after they are defined, but other changes made to the screen or fields during the current edit process are not. The CANCEL key on the 2236DW or 2336DW terminal is not operational as the CANCEL key in IDEAS. IDEAS always uses FN 31 for CANCEL to avoid confusion with EDIT.

FN/TAB

Use when defining a subscreen to indicate areas of the screen that will not overwrite the previous screen. The FN key creates a blank (nondistructive, null) at a given character position, rather than a space (a destructive space). The Screen Mask editor displays a null as a blank and a space as a dot. Although you can specify blanks (or nulls) on any screen, they have no meaning except on subscreens.

A subscreen does not issue a "clear screen" when loaded, as do all other screens. When defining a subscreen, use the FN key to specify any character positions that should not be overwritten when the subscreen is loaded. A chained screen does not erase boxes generated by the main screen; to erase boxes, you must use the Pass/Fail Option H, described in Section 5.5.

SHIFT FN/SHIFT TAB

Sets all character positions on the current row, starting at the current column and continuing to the end of the line (except fields), to blanks (nulls).

RETURN

Moves the cursor to Column 1 of the next row down unless the cursor is on Row 23, in which case the cursor goes to Column 1 and the alarm sounds.

RUN/EXECUTE

Saves the screen mask and fields on the currently specified disk and reloads the Screen Utilities menu. This is one of the few times throughout the IDEAS utilities where you cannot use RETURN interchangeably with EXECUTE.

BACKSPACE

Acts as a destructive backspace and replaces any null or text character with a space (dot). If the cursor at Column 1 or the cursor position to the left of cursor is a field position, the alarm sounds and the cursor remains in place.

4.1.3 Field Definition

You must define each field on a screen to control its subsequent use, appearance and functionality. The following procedures provide access to field definition.

To define a new field, perform the following actions:

1. Position the cursor at the screen location for the first character of the field you wish to create. You must be sure that no portion of this field will overlap additional text or fields on the same line.

2. Press FN 00. The Field Definition screen (Figure 4-4) appears. The field name for the first new field defaults to FIELD001 and the field number for the first new field defaults to 1. Each subsequent field name and field number defaults to the next available number (FIELD002, Number 2; FIELD003, Number 3; etc.). You can change these names as you wish. The length defaults to 0 characters and must be changed before you complete field definition or enter Edit Specifications.

To edit a defined field, perform the following actions:

1. Position the cursor under any pseudoblank (block) within the field you wish to edit.

2. Press FN 00. All attributes appear as previously defined.

Work Fields

When defining fields on a screen, you are not limited to fields associated with data files. In fact, when defining the Edit Specifications for a screen, you may often find it necessary to use unassociated "work fields" to store information of various sorts. Like any other field on a screen, you can designate a work field as displayable or nondisplayable.

Field At Row 24, Column 80

Fields defined at Row 24, Column 80 are special fields, 1 byte in length, used for input via function keys, EDIT, EXECUTE, and FN. When you define and edit this field, most attribute values default and cannot be modified. You can only modify the field name and its position in the work buffer. You can also specify Edit Specifications at this field, as described in Chapter 5.

You cannot specify anything else on Line 24; you cannot place a text character in this position (24, 80). If you move the cursor to this position, the only operations you can perform are the following.

- Define or edit the special field (FN 00).
- Move the cursor up (FN 05 or 06, or NORTH).
- Move the cursor to the beginning of the screen (FN 07).

When you touch FN 00 to define a field, the Field Definition screen, Figure 4-4, appears.

```
IDEAS Screen Mask MAILSCRN - Field Attribute Specifications       Release 2.1

  FN ATTRIBUTE                          FN ATTRIBUTE
  00 File number    0                   07 Right Justify              N
     Data files    4=                   08 Zero fill                  N
     1=MAILLIST    5=                   09 Decimals                   0
     2=            6=                   10 Display option             2
     3=            7=                      0=Not displayed  2=Bright
  01 Field               FIELD002          1=Normal         3=Blink
     Field number               2       11 Keyboard entry             Y
  02 Row                        7       12 Required field             N
  03 Column                    23       13 Full if present            N
  04 Length                     0       14 EXEC required              N
  05 Position                  80       15 Non-bypassable             N
  06 Character Type             7       16 Number of edits            0

  Valid Character Types                 FN Processing Option
  0=Digits       5=U/C & Digits         16 Access edit specifications
  1=Digits + -   6=U/C num punct        24 Copy all edits
  2=Digits & .   7=Any Character        25 Delete this field
  3=Any Numeric  8=Yy1=1,NnO=0          26 Insert as new field
  4=U/C letters  9=EXEC EDIT FN'        31 Cancel field edit
                                        FN (TAB) Display names
Touch the FN key corresponding to the desired operation or EXECUTE to accept
```

Figure 4-4. Field Definition Screen

4.1.4 Field Definition Options

An explanation of the options of the Field Definition screen follows. When you define a field associated with a date file, certain attributes are supplied automatically and cannot be changed. They are the following: LENGTH, POSITION, VALID CHARACTERS, RIGHT JUSTIFY, ZERO FILL, and DECIMALS. To change these attributes, return to the Data File Specification module. (Refer to Section 3.2.)

It is important to note that all logical processing is established by defining Edit Specifications for the field. To do this, touch FN 16 from this screen, as described below. Chapter 5 presents instructions for defining Edit Specifications.

FN 00 -- FILE NUMBER

If the field you are defining is from an associated data file, enter the number of that data file (the associated data files are listed on the screen beneath this entry). If no associated data file contains this field, enter zero as the file number. Only zero and valid file numbers are appropriate entries.

This response, in conjunction with FN 01 (FIELD), indicates that input to or manipulation of this field occurs directly in the work buffer location assigned to the field. Any field that is part of an associated data file uses the previous content of the record as a default value, if available. Any changes made to fields are automatically saved when the record currently being processed is saved.

FN 01 -- FIELD

The field name should be any eight-character mnemonic used to identify this field on the screen. Each field name specified must be unique to the screen. If the field is associated with a data file, the field name must be a currently defined data file field name. The screen does not accept invalid field names.

The IDEAS utilities use some specific field names that you should not use. These names, some of which are described in Chapter 5, are @TSTFLDn, @SYSBUF0, BLANKFLD, CPU ID #, STATION#, TERMINAL, PARTIT'N, OPERATOR, USERCLAS, NEWTRAN#, NEXTSEQ#, SYSFLAG#.

The field number indicates the sequential order of field processing. IDEAS automatically defaults the field number to the next available number. The position for fields not associated with a data file is supplied automatically as the next available position in the screen work buffer. To change the field number of a defined field, use FN 25 (described below), which is the "delete field" option. After you delete the field, the screen prompts offer the option of inserting the field into a new position. To insert a new field into a defined order, use FN 26 (also described below).

FN 02 -- ROW

This attribute indicates the screen row on which the field is to exist; it defaults to the current cursor position. Valid input is 1 to 23.

FN 03 -- COLUMN

This attribute indicates the screen column on which the field is to exist; it defaults to the current cursor position. Valid input is 1 to 80 and is verified with the LENGTH to ensure that the entire field will fit on the screen.

FN 04 -- LENGTH

All fields must have a length of at least 1 character and must completely fit between the current column and column position 80. For numeric fields, remember to include one character position each for a sign and/or decimal point, if used (for example, -999.99 has a length of 7).

You cannot define a field that will overlap an existing field. If you attempt to do this, an error message appears at the bottom of the screen indicating that you should modify the field attributes. Defining a field that overlaps existing screen text overwrites that text.

You can define fields of up to 255 bytes to use as nondisplayable, non-keyboard-entry work fields to use when defining Edit Specifications. To do this, you must use FN 10 to indicate that the field is not displayed and FN 11 to indicate that keyboard entry is not allowed. Fields defined in this manner take up only one byte on the screen mask, regardless of field length. Refer to Subsection 5.2.13 for further information on work fields.

FN 05 -- POSITION

This indicates the starting position within the screen work buffer of the field being accessed.

If you will access data fields by both a screen and its subscreen, you must align buffer positions of all chained screens. In effect, you must define all data files on the main screen in the same order as on its subscreen, even if one screen does not access all the data files the other screen does. You can redefine work fields at identical buffer positions if information is shared, or at mutually exclusive positions if not.

FN 06 -- VALID CHARACTERS

You can limit the valid character set of a field to one of nine subsets of the keyboard. The system checks valid numeric conventions on input and signals invalid entries audibly. If a field is defined as uppercase only, the system automatically translates lowercase to upper case characters at data entry. A description of the different field types follows.

Input	Subset Type	Input Characters Allowed
0	Digits	0 through 9, (Numeric)
1	Digits +, -	0 through 9, + (Plus), - (Minus)
2	Digits .	0 through 9, . (Decimal point)
3	Any Numeric	0 through 9, +, -, .
4	U/C Letters	Uppercase (A through Z)
5	U/C Digits	Uppercase (A through Z), 0 through 9
6	U/C Num Punct	As in number 5 and punctuation characters: HEX(20) to HEX(5F)
7	Any character	As in number 6 and lower-case letters: HEX(20) to HEX(7A)
8	Yy1=1, Nn0=0	Allows only input of the characters indicated. These are stored in the work buffer as input.
9	EXEC EDIT FN	Allows only input of the EXECUTE key, EDIT key, function keys 00 through 31, the FN key, or the shifted FN key

FN 07 -- RIGHT-JUSTIFY
A Y response right-justifies display of the field on the screen regardless of input size or type. An N response implies left justification.

FN 08 -- ZERO FILL
A Y response fills blank spaces to the left of the first input character with numeric zeros (0). This attribute only applies to character types 0 through 3 (numeric).

FN 09 -- DECIMALS
This field requires a numeric response indicating the number of decimal places allowed for input or displayed numeric fields. You can set this field only if you set the field to Character Type 2 or 3.

FN 10 -- DISPLAY OPTIONS
These indicate how the field should appear when displayed on the screen. This attribute is also used to indicate that fields are nondisplayable. The valid responses are the following:

- 0 Not Displayed
- 1 Normal
- 2 Bright
- 3 Blink

FN 11 -- KEYBOARD ENTRY
A Y response indicates that processing should stop at this field to allow input.

FN 12 -- REQUIRED FIELD
A Y response indicates that some input must be made to this field from the character type indicated in FN 06. Blank operator responses are not be permitted.

FN 13 -- FULL IF PRESENT
A Y response indicates that if you make any entry into the field, you must fill all character positions before the system will accept the field. You can use this attribute for fields containing phone numbers or zip codes, where you might not require field entry, but where the field, if entered, should be full.

FN 14 -- EXEC REQUIRED
A Y response means that you must terminate entry to the field with an EXECUTE or RUN keystroke. An N response means that after you fill the last character position within the field, the field terminates automatically and control passes to the next field. You can terminate a partially filled field with an EXECUTE keystroke if it is not a FULL IF PRESENT field.

FN 15 -- NON BYPASSABLE
A Y response in this field ensures that the operator cannot skip this field using skip-ahead keys. A field at 24, 80 defaults to a Y response.

FN 16 -- ACCESS EDIT SPECIFICATIONS
This option, which displays the current number of Edit Specifications associated with this field, defines all field processing for the current field. Touching FN 16 evokes the Edit Operation Sequence Selection screen (Figure 5-1). Edit Specifications are discussed in Chapter 5.

You can generate some standard simple programs without specifying the processing logic when defining the associated screen. You can use this feature of IDEAS to create a simple data entry, random inquiry, sequential inquiry, or add/change/delete program quickly. Refer to Section 9.3 for details.

Other functions that can be performed on the field follow.

FN 24 -- COPY ALL EDITS
You can use this function to copy all Edit Specifications currently in the Edit Specification file from any field on any screen to the current field. The screen and field names you specify change to the current screen and field names during copying. You must change other associated screen and field references denoted in the Edit Specifications manually as needed.

A field can have many Edit Specifications associated with it, and you may wish to copy only a particular processing sequence from a field on another screen. You can do this with the Edit Specification Auto Recall capability. Refer to Section 5.2.

The Edit Specification file is separate from the Screen Mask file. Because of this, you can copy edits from a screen even if you deleted the screen mask itself from the system, as long as you did not delete its associated edits from the Edit Specification file. With this capability, you can establish standard sequences of Edit Specification operations (a "library" of edits).

FN 25 -- DELETE THIS FIELD
During field editing, this option allows you to delete the current field or reinsert the field with a new sequence number by changing the FIELD NUMBER when requested.

FN 26 -- INSERT AS NEW FIELD
During initial field definition only, this option allows you to order the current field to be executed before previously defined fields. To do this, enter the desired FIELD NUMBER. The numbers of the previously defined fields change to accommodate the insertion of the new field.

FN 31 -- CANCEL FIELD EDIT
You can cancel current processing at any time during field specification. Control returns to screen specification. If the field existed previously, the values revert to the condition prior to the field edit. If you defined or modified Edit Specifications during the field specification, these changes or modifications remain on record after a cancel from field specification.

FN (TAB) -- DISPLAY NAMES
To view the names of the fields defined for each associated data file, touch the FN key on the 2236DE or the TAB key on the 2236DW. You can display data file field names using the appropriate listed FN key. Field names for the screen are those present when editing of the screen began; this option does not reflect any fields created during the present edit cycle.

4.2 ADDITIONAL SCREEN UTILITIES

The screen utilities also revise and document existing screens, document existing screens in batch mode, and file and copy screens.

4.2.1 Screen Revision

The Screen Mask Revision module (FN 01 from the Screen Utilities menu) revises an existing screen mask and its associated fields and Edit Specifications. Enter the name of the screen to be revised, and proceed as described in Section 4.1.

If you change the names or the order of the associated data files when revising a screen, you must revise all fields on the screen mask that reference the changed data files to update the file number and field name. You must also revise any operations that reference fields in the modified data files.

4.2.2 Screen Documentation

When you choose the Document option (FN 02) from the Screen Utilities menu, a screen appears requesting the name of the screen to be documented. After you specify a name, a message appears asking whether to attach the field edits to the documentation. An answer of N indicates that you will produce the screen mask and the attributes of its fields. This saves both time and paper if you want only the screen mask. An answer of Y appends a listing of the Edit Specifications associated with the fields to the documentation.

The Screen Mask Specification screen (Figure 4-3) for the screen to be documented appears. If the screen is the correct one, accept the screen and print the documentation; otherwise, touch CANCEL (FN 31) to return to the menu.

4.2.3 Extra Screen Utilities

When you touch FN 03 the Supplementary Screen Utilities menu (Figure 4-5) appears.

```
IDEAS Screen Mask Utilities - Supplementary Utilities Menu          Release 2.1


                                 FN Program Module
                               ■ 00 Screen filing
                                    Copy to/from archive
                                    Delete edits/screens
                               · 03 Copy screen
                               · 04 Batch screen update


SEP 27 83                                                           Term #  2
Mem 36.00                      Global Selected                      Sta. # 20
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 4-5. Supplementary Screen Utilities Menu

Screen Mask Filing

When you touch FN 00 from the Supplementary Screen Utilities menu, the Screen Mask Filing screen (Figure 4-6) appears. When you file a screen, it is stored in an archive file that includes its Edit Specifications (unless the Edit Specifications were deleted from the Edit Specification file previously). Thus, screen filing performs two functions: it enables you to clear room in the Edit Specification file by storing the specifications associated with a particular screen, and it provides a means of screen backup. The screen archive is of variable size, depending on the amount of information in the screen file and the number of edits.

Once you archive a screen, you can still use the archived file in running an application by specifying the location of the archived screen as the location of screen files on the Peripheral Device Selection screen (Figure 2-6). You need not archive a screen along with its Edit Specification in order to use it in running an application; however, you cannot revise or document the screen unless you retrieve it from archive. Using this feature, you can compress a screen for a delivered application by archiving it after you delete its edits.

```
IDEAS Screen Mask Filing Operations                              Release 2.1

Please select      Screen Mask Filing Operations Available
one of the 5       ------------------------------------
operations      1. Archive screen and edits, delete edits from main edit file
shown in the    2. Archive screen and edits, preserve edits in main edit file
list at right   3. Retrieve all screen data from archive
Operation = 2   4. Delete edits from main edit file, preserve mask & fields
Archive =       5. Delete entire screen, including edits

Note:  Main edit file is currently  77 % full with  498 empty slots remaining.

First screen to process              One screen may be processed, or a range of
Last screen to process               screen names.  Selection may be specified
Version                              any combination of version, application,
Application                          function, and user ID (all AND selection.)
Function                             If user ID selection is used, it may only
Create/revise user ID                be the current user's ID.
```

Figure 4-6. Screen Mask Filing Screen

As indicated on the screen, there are five filing options.

1. Option 1 allows you to copy a screen into an archive and delete its associated Edit Specifications from the Edit Specification file. After doing this, you can still access the screen using the Screen Revision utility, but the screen will have no associated Edit Specifications. In addition, a new screen cannot copy any of the Edit Specifications associated with the archived screen, and the Program Generation utility cannot access the special edits for the archived screen. Any program previously generated for that screen continues to work as specified, because it incorporates the edits.

2. Option 2 copies a screen to an archive but leaves the screen and its associated Edit Specifications intact. You can use this option to back up existing screen specifications while allowing further revision of the screen.

3. Option 3 restores a screen by retrieving the screen and its associated Edit Specifications from an archive and placing the Edit Specifications in the Edit Specification file.

4. Option 4 leaves the screen and its fields intact, but eliminates all associated Edit Specifications from the Edit Specification file.

5. Option 5 deletes the screen and its edits. Once you exercise Option 5, you can no longer access the screen unless the screen was archived previously. In this case, you can retrieve the screen using Option 3.

If you choose Option 1, 2, or 3, the screen requires the disk address of the archive disk. This address must be one of the addresses specified on the Peripheral Device Selection screen, Figure 2-6.

As indicated in the lower left corner of the screen, you must indicate which screen or screens to process. To process only one screen, specify that screen as both the first and last screen to process. To process more than one screen, specify a first and last screen. The utility then searches the disk index and performs the indicated operation on all IDEAS Release 2 screens alphabetically greater than or equal to the first screen and less than or equal to the second screen. If you leave both screens blank, then the utility processes all IDEAS Release 2 screens. If you leave the second screen blank, then the utility processes all screens greater than the first screen. Similarly, if you leave the first screen blank, then the utility processes all screens less than the second screen.

You can add additional limitations to the number of screens being processed by selecting only screens of a particular Version, Application, and/or Function to process. You can also process only screens to which the current user ID has Creation/Revision privileges; you cannot specify another individual's user ID.

Copy Screen

The Copy Screen utility makes a copy of an IDEAS screen under a different name. You can copy a screen with or without its associated Edit Specifications. When you touch FN 03 from the Supplementary Screen Utilities menu, the screen requests the name of the mask to be copied and asks whether to copy the associated Edit Specifications along with the screen mask. You must then enter the name of the new screen. If the name of the new screen is the name of a previously defined IDEAS screen, a warning message appears: you can touch EXECUTE to overwrite the previous screen with the copied screen, or you can touch EDIT to change the name of the new screen. After you accept the screen, the Supplementary Screen Utilities menu (Figure 4-5) appears. If you wish to edit the copied screen, you must use the Screen Revision utilities.

Batch Screen Update

Editing a data file and changing the length of any field in it can cause errors, since the screen work buffer positions become incorrect. Similarly, changing other attributes of the data file fields can cause errors in a screen that references the data file. Batch Screen Update corrects for these errors. Note that you cannot use this option on archived screens because archived screens cannot be revised without first retrieving them from archive.

Batch Screen Update processes one or more screen files on disk and compares the field attributes in them with the associated data files containing the same fields. If specifications for data file fields have changed, the utility copies the changes into the screen field attributes. The attributes that are compared and updated are: zero fill, justify, decimal, character type, length, and position in screen buffer. The system determines the buffer position by adding the field's data file position to the total length of the preceding data files.

When you choose the Batch Screen Update option from the Extra Screen Utilities menu, the Batch Screen Update screen (Figure 4-7) appears.

```
IDEAS Screen Mask Filing Operations                                    Release 2.1

                          Batch Screen Update Specifications
                          ----------------------------------------

  Now processing:  Index sector            out of            sectors

                   Screen name
                   Number of screens updated


First screen to process ********    One screen may be processed, or a range of
Last screen to process              screen names.Selection may be specified by
Version                             any combination of version, application,
Application                         function, and user ID (all AND selection.)
Function                            If user ID selection is used, it may only
Create/revise user ID               be the current user's ID.

Option: Work fields on the specified screens may be moved to available free
space in the screen buffer (Y), or listed on a printout (N)
```

Figure 4-7. Batch Screen Update Screen

If you leave the First screen to process field blank, processing begins at the lowest possible value for a screen name. If you leave the Last screen to process field blank, then the program searches for and processes only the screen entered as First screen to process. If you leave both fields blank, then all screens are processed.

Specifying Version, Application, and Function limits processing to a range of screens belonging to the desired sets. Entering a user ID, which must be that of the current user, restricts processing to screens naming that ID as having exclusive edit and documentation privileges.

The entry screen asks whether you want the program to shift work fields (those unassociated with a data file) to available free space in the screen buffer, or to list them on a printout. This option is given because updating a screen according to the attributes of a data file may cause shifted data file fields to overwrite the positions of work fields in the screen buffer. The Update Screen utility can calculate the offset to avoid this, and can assign new positions to work fields. If the space in the buffer following the data file records is not sufficient, the work fields are listed in a printout, even if you did not specify the printout option. You can refer to the printout to edit the screen's listed work fields individually and place them in safer positions.

4.2.4 Batch Screen Documentation

When you choose the Document (Batch) Utility (Option 4 on the Screen Utilities menu), the Screen Mask Batch Documentation screen (Figure 4-8) appears. You can use the Batch Screen Documentation utility to document a series of IDEAS screens at the same time.

```
IDEAS Screen Mask Editor - Screen Mask Batch Documentation Module  Release 2.1

Note:  This module will document all screen mask
       files which meet the following criteria:

    1. A password is not required for documentation
    2. The edit/document user class is <= yours
    3. The user ID, if any, is yours
    4. The version, application code, and function code
       are as specified below.  A blank entry in any of
       these 3 will cause that parameter to be ignored.

Version            =

Application code   =

Function code      =
```

Figure 4-8. Screen Mask Batch Documentation Module

You must indicate any Version, Application, or Function parameters and accept the screen to initiate documentation.

CHAPTER 5
SCREEN EDIT SPECIFICATIONS

5.1 OVERVIEW

The Edit Specifications section of the Screen Definition utility serves as the entry into the section of the IDEAS utilities through which you define all processing. The field edits and their associated options form what amounts to a programming language. Through this "language," you determine the flow of control for the data processing by indicating what conditional actions to perform under what circumstances. The variety of actions that IDEAS supports is extensive and enables you to write most applications without writing any BASIC code.

For each field on the screen, you can designate up to 32 edit operations to perform before data entry and up to 32 edit operations to perform after data entry. Each edit operation, however, can include up to 50 actions, and a subsequent edit operation can continue the previous operation. You implement all "programming logic" through these actions.

Edit operations in an IDEAS application set a pass or fail condition, and the pass/fail state determines subsequent actions. Each IDEAS screen that defines an edit operation leads to the Pass/Fail Action Specification screen (Figure 5-11), which lists 26 types of actions and tests (labeled A through Z) you can specify. The following text refers to Pass/Fail Option #, where # is a letter indicating which pass/fail option is appropriate. Section 5.4 describes the Pass/Fail Action Specification screen and associated options.

As you define screen processing, IDEAS stores it in an Edit Specification file (established during system installation; refer to Section 1.5). The Program Generation utility, described in Chapter 9, uses the Edit Specifications associated with a screen when generating the program associated with that screen. You can archive a screen along with its Edit Specifications, thereby freeing space in the Edit Specifications file. Subsection 4.2.3 presents instructions for archiving screens. This Edit Specifications file is shared by reports, described in Chapter 6, and batch processing programs, described in Chapter 10.

Before you define the Edit Specifications associated with a screen, plan out the data entry program. You should determine such things as at what point you wish to read a record or save a record, what logical tests to perform, and when to perform the logical tests. A description of some of the capabilities and features of the Edit Specifications follows. This will familiarize you with the tools with which you will work. After these general explanations, a screen-by-screen description gives instructions for the implementation of the Edit Specifications options.

You can generate some standard simple programs without specifying the processing logic when defining the associated screen. This feature of IDEAS creates a simple data entry, random inquiry, sequential inquiry, or add/change/delete program quickly. Refer to Section 8.2 for details.

5.2 SPECIAL FEATURES

Edit Specifications provide many features that enhance the versatility of an IDEAS-developed application. Some of these features are important for even simple applications, while others are necessary only for more complex applications. An overview of these Edit Specification features follows. Some of the more frequently used features are described first: defaults, dates, chaining screens, screen messages, Edit Specification auto recall, and library screens. Then system flags and the use of extra conditions and operations are summarized; these two features are important, although it is possible to develop an application without them, particularly if you use the library screens. Finally, some of the more advanced features, which most basic applications do not require, are presented: counters, trapping function keys, @TSTFLDs, and the system buffer.

5.2.1 Defaults

You can specify that a field on an IDEAS screen default to a defined value. This default can be any of a number of things, including another field on the screen, the screen name, the CPU number, the terminal number, the operator ID, the user's security class, and @TSTFLD. To assign a default value to a field, use Pass/Fail Option I. This option copies the contents of one field to another field. You can use this pass/fail specification with the Logical Test Edit Specification so that assigning a default value to a field is conditional, depending on whether the field is blank. This pass/fail option gives a complete list of available default variables.

5.2.2 Dates

Pass/Fail Option Z tests for a particular date format. This option also can supply the current system date or the most recently entered date as a default value. Refer to the explanation of Option Z.

5.2.3 Chaining Screens

You can specify, as part of a particular screen's processing, that another screen be loaded and displayed. You do this by using Pass/Fail Option S or T and loading in a program module to load a new screen. The new screen can be a screen defined as a subscreen. This function allows you to create applications that consist of a virtually unlimited sequence of data entry screens. Refer to the explanation of Option S for details.

5.2.4 Screen Messages

You can specify up to nine 64-character messages per edit as operator prompts or error messages for use in conjunction with any of the pass/fail actions. To define the messages, touch FN 16 from the Pass/Fail Action Specification screen. Pass/Fail Option H displays the messages. Except when displayed on Line 24, the messages appear at normal intensity, even if displayed on the first line of a screen where that line of text is bright. Refer to the explanation of Option H for details. The messages that you define are in addition to the IDEAS-supplied messages described in Subsection 5.2.14.

5.2.5 Edit Specification Auto Recall

You can copy an individual sequence of Edit Specifications from a field on any screen onto the current screen as long as it is stored in the Edit Specification file. This avoids much duplication of effort. Instructions for implementing this option follow Figure 5-2.

5.2.6 Library Screens

A library of simple screens set up for reference purposes provides the ability to create simple add, change, or delete programs for screens. You can recall the Edit Specifications for each field on these screens using the Edit Specification Auto Recall function. The IDEAS software includes the following screens, in archived form:

- IDS2sADD -- Add a record to a data file
- IDS2sINQ -- Inquiry into a data file
- IDS2sSEQ -- Find the first and next logical record in a data file
- IDS2sACS -- Add, change, or delete records in a data file without checking for duplicate keys on key field entry
- IDS2sACD -- Add, change, or delete records in a data file that checks for duplicate keys on key field entry

Wang supplies the library screens, which are documented in Appendix B. However, the installation utility does not retrieve them automatically. You must use the Screen Mask Filing option of the Supplementary Screen utilities (described in Subsection 4.2.3) to retrieve a library screen from archive. After you retrieve a screen, the Edit Specification file on the system disk contains its associated Edit Specifications. Which disk you store the screen files on does not matter, since IDEAS retrieves the edits from the Edit Specification file. If the IDEAS software arrived on two DSDD diskettes, the library screens are on the second diskette. If the software arrived on SSSD diskettes, the screens are on the sixth diskette. You can establish your own collection of library screens.

5.2.7 System Flags

Thirty-six system flags are available to the developer. These flags are called SYSFLAG#, where # can be a digit (0 through 9) or an uppercase letter (A through Z). The system flags are variables which have two states: ON and OFF. As part of a defined edit operation, you can turn the system flags on or off individually or test them for their on/off status. Loading any menu module sets all flags to the off position.

The library screen used in the familiarization exercise provides an example of system flag use. In this example, System Flag D is turned on if a record is found after a record read. This distinguishes between a record that is not on file and a record that is on file but is being used by another partition, as described under Step 34. Also in the example used in the familiarization exercise, System Flag G notes that the operator is editing the information currently displayed and not reading a new record, as described under Step 40.

System flags specify conditional actions, as explained in Section 5.6.

The Report utility imposes some special conventions concerning System Flags 1 through 7 and System Flags A through G. Section 6.3 describes these conventions. Although these conventions are neither automatic nor necessary when running the Screen utilities, you may find it useful to follow them when you specify edits on a screen.

5.2.8 Conditions and Operations

When you define a pass/fail operation on the Pass/Fail Action Specification screen, you are presented with two 8-character fields next to the Action Specification Field labeled Operand 1 and Operand 2. These fields generally specify the field names, file names, and screen names required to complete the action specification. Options H, O, P, Q, R, S, and T, however, do not always require that you use both fields, and Options A, B, C, D, E, F, G, and blank do not always require that you use any field. In these cases, the empty field(s) can specify one of several additional operations and/or conditions.

You can define these additional operations and conditions through the use of certain keywords that serve as a type of programming statement. These keywords perform such functions as testing the current state of a system flag or the value of the last function key pressed. The specified pass/fail action is performed only if the conditions are met. Other keywords can perform such operations as changing the state of a system flag or branching to a different location in the pass/fail sequence. Operations and conditions and their associated keywords are described in detail following the explanation of the pass/fail options in Section 5.5.

5.2.9 Counters

IDEAS supplies two counters as default fields: the fields NEXTSEQ# and NEWTRAN#. NEXTSEQ# is a 3-digit sequence number unique to each partition. It increases incrementally each time it is used, and reverts to zero when it exceeds 255, when a menu is loaded, or when specified by one of the extra pass/fail operations described in Section 5.6. NEWTRAN# is an 8-digit number that IDEAS reads from disk, supplies to the program, increases incrementally, and saves back to disk each time an IDEAS program uses it. It is common to all partitions and/or CPUs accessing its disk. When it exceeds 99999999, it reverts to 0. During runtime, if an operator cancels out of a screen that evoked NEWTRAN# without saving a record, the counter still increases incrementally, leaving a gap in its sequence. The developer can supply the current counter as a default by using Pass/Fail Option I.

5.2.10 Function Keys

You can define function keys at both the field level and the screen level. As described in Subsection 4.1.1, you can indicate up to eight screen-level function keys. When you enter any of these function keys at runtime, processing continues with the pre-entry special edits on the last field on the screen. At this edit sequence, you can define conditional operations that depend upon the last function key pressed. Refer to Section 5.6, Conditions and Operations.

You can also enable function keys at the field level using Pass/Fail Option Y. This option allows the IDEAS-generated program to accept designated function keys as valid input, and you can define conditional operations according to the function key pressed. Refer to Section 5.6.

Field-level function keys take precedence over screen-level function keys: a field for which field-level function keys are defined does not execute a screen-level trap for the same key. Similarly, screen-level function keys take precedence over any pre-set use of function keys. For example, if you trap FN 31 at the screen level, touching FN 31 continues processing with pre-entry for the last field on the screen rather than cancelling out of the screen.

5.2.11 @TSTFLDs

IDEAS-2 supports ten partition-related special fields called @TSTFLD0 through @TSTFLD9. These fields are unformatted variables that you can use, access, modify, and reference without having to specify them as screen fields. They are each 256 bytes in length, and you can use them interchangeably with screen-specified fields at any time. @TSTFLDs do not take up any space in the work buffer and their values remain unchanged from menu to menu when new screens or programs are loaded. START modules do not clear @TSTFLDs.

@TSTFLDs are normally stored in a data file on the IDEAS system disk that is exclusive to a particular partition of a particular CPU. Each utitities disk has its own set of @TSTFLDs. Each reference to @TSTFLDs requires a disk access to recover this data; therefore, you should consider the use of @TSTFLDs in relation to increased performance time and disk use. Note the existence of @SYSBUF0, described below, and screen work fields, described in Subsection 4.1.3.

5.2.12 System Buffer

IDEAS supplies a single System Buffer, @SYSBUF0, initially set or cleared by START modules, that you can use wherever you require screen fields. Since this field is stored in foreground partition memory, you need not define it on the screen to access it. The system buffer remains from menu to menu. @SYSBUF0 is identical in function to @TSTFLD#, but @TSTFLD# is 256 bytes long and requires a disk read to access it, while @SYSBUF0 is 64 characters long and does not require a disk read. The System Buffer is not the same thing as the screen work buffer, whose size is determined during the START program generation module.

5.2.13 Work Fields

As Subsection 4.1.4 describes, you can define fields on the screen mask that cannot be displayed on the screen and cannot be entered from the keyboard. These fields can be up to 255 bytes long, although they only take up 1 byte on the screen mask. You can use these fields whenever you need a work field in defining Edit Specification operations.

5.2.14 System Prompts and Error Messages

The IDEAS system provides its own set of error messages and prompts. These error messages are appear automatically as necessary during an application. You can use P/F Option G to disable these error messages.

Each system error message has a specific error message number. You can test for this message number and base subsequent processing on it. The procedure for testing for the system error message number is described in Subsection 5.6.1.

Appendix H contains a list of the system error messages and their numbers.

5.3 EDIT OPERATIONS

When you touch FN 16, the Edit Operation Sequence Selection screen appears (Figure 5-1).

```
IDEAS Screen Mask Editor          Edit Operation Sequence Selection   Release 2.1
Note: Edit specification file is 77 % full.   511 records may be added.
For screens only, 1-32 are pre-entry and 33-64 are post-entry.  For report and
batch there is is no pre/post distinction.  To select item to be edited, move
the cursor to the desired number using FN 5,6,12,&13 (or N,S,E,W), then "EDIT"
              Pre-entry Operations                     Post-entry Operations
No Description        No Description          No Description        No Description
01 - available -      17 - available -        33 - available -      49 - available -
02 - available -      18 - available -        34 - available -      50 - available -
03 - available -      19 - available -        35 - available -      51 - available -
04 - available -      20 - available -        36 - available -      52 - available -
05 - available -      21 - available -        37 - available -      53 - available -
06 - available -      22 - available -        38 - available -      54 - available -
07 - available -      23 - available -        39 - available -      55 - available -
08 - available -      24 - available -        40 - available -      56 - available -
09 - available -      25 - available -        41 - available -      57 - available -
10 - available -      26 - available -        42 - available -      58 - available -
11 - available -      27 - available -        43 - available -      59 - available -
12 - available -      28 - available -        44 - available -      60 - available -
13 - available -      29 - available -        45 - available -      61 - available -
14 - available -      30 - available -        46 - available -      62 - available -
15 - available -      31 - available -        47 - available -      63 - available -
16 - available -      32 - available -        48 - available -      64 - available -
Attention:      Touch EXECUTE to return to field attribute specifications
```

Figure 5-1. Edit Operation Sequence Selection Screen

After you move the cursor to the desired operation number and touch EDIT, the Edit Operation Type Sequence Selection screen (Figure 5-2) appears. You cannot change the sequence number of an operation, although you can copy individual edits. You should leave gaps between defined operations to allow room to insert additional operations later.

```
IDEAS Screen Mask Editor        Edit Operation Type Selection        Release 2.1

        FN Type of operation
        -- ---------------------------------------------------------

        00 Perform pass/fail actions only using last P/F condition

        01 Set field(s) equal to field(s) and/or constant(s)

        02 Read a record from a data file

        03 Perform logical test(s)

        04 Math calculation(s)

        05 Range test(s)

        06 Table look-up or table look-up and replace

        07 User-supplied process

Attention:      Select operation using FN'00-07 above or FN'31 to cancel
```

Figure 5-2. Edit Operation Type Selection Screen

IDEAS supports seven types of operations. You can perform each operation on any field defined on the screen, singly or in combination. Any of the operations can result in a pass or fail condition. Based on the pass or fail result for each test, you can use pass/fail specifications to specify that certain actions are to occur in the applications program.

For each type of operation, touching the appropriate function key evokes the screen on which you indicate whether the operation is conditional based on a system flag and the parameters of the operation. The completion of this screen, for each option, evokes the Pass/Fail Action Specification screen (Figure 5-11), discussed in Section 5.4.

You can recall a previously-defined Edit Specification sequence on any operation screen by performing the following steps:

1. When the screen first appears, the cursor is at the field labeled "Conditional?". Use the skip-back keys (FN 13 or 14) to move the cursor to the field labeled "Screen" in the upper left corner.

2. Enter the name of the screen and field from which you want to copy the Edit Specification sequence, and enter the number of the sequence you wish to copy (as numbered on the Edit Operation Sequence Selection screen, Figure 5-1). Entering 00 for sequence number recalls the first edit for the currently specified operation type. For example, if you are defining a math operation, and you recall a screen and field but indicate 00 as the sequence number, IDEAS copies the first math operation defined for that field.

3. After you accept the sequence number, the cursor moves to the next field, and the three fields you entered revert to the current screen, field, and sequence number. After you accept the screen, the Pass/Fail Action Specification screen appears. The recalled Edit Specification sequence should appear on the screen.

As noted in the upper left corner of each screen, any action can be dependent upon the state of a system flag. You can set a system flag as part of a pass/fail sequence, as described in Section 5.6.

5.3.1 Perform Pass/Fail Actions Only

When you select FN 00 from the Edit Operation Type Selection screen (Figure 5-2), the Perform Pass/Fails Only screen (Figure 5-3) appears.

```
IDEAS Screen Mask Editor        No-op - Perform Pass/Fails Only     Release 2.1

Screen      MAILSCRN    When a no-op is used, there is no operation performed to
Field       CUST #      establish a new pass/fail condition.  Pass/fail actions
Sequence number 01      specified for this edit will be performed based on the
Conditional Y/N? N      currently existing pass/fail condition established by a
Only if SYSFLAG =       prior entry, edit, or operation only.

A no-op may be used in a number of situations:

1. If used immediately following a keyboard entry, the pass/fail condition is
   set by the validity of the entry.  For example, entry of an invalid date,
   when a date edit is specified will cause a "fail" condition to exist.

2. If used immediately following another edit type (ie. range, table, etc.),
   it extends that edit's number of pass/fail actions beyond the normal 50
   because the pass/fail is not changed, and this becomes merely an extension.

3. If used merely to specify certain activities, the P/F code should be "B"
   (both) because the currently existing pass/fail condition is indeterminate
   since a no-op does not set the pass/fail condition.
```

Figure 5-3. Perform Pass/Fails Only Screen

This operation is a "no-op"; you perform no operation to establish a new pass/fail condition. Pass/fail actions specified for this edit are performed based on the currently existing pass/fail condition established by a prior entry, edit, or operation only. If you specify this as the first edit for a field, indicate B (both) for the pass or fail conditions defined on the next screen (the Pass/Fail Action Specification screen, Figure 5-11).

After you accept this screen, the Pass/Fail Action Specification screen (Figure 5-11) appears. Refer to Section 5.4.

5.3.2 Set Field(s) Equal To Field(s) And/Or Constant

When you select FN 01 from the Edit Operation Type Selection screen (Figure 5-2), the Set Field Equal to a Constant Operation screen (Figure 5-4) appears.

```
IDEAS Screen Mask Editor        Set Field(s)=Field(s)/Constant(s)  Release 2.1

Screen      MAILSCRN    The F/C column below is used to specify if the component
Field       CUST #      is a field name (to use the field value) or a constant.
SEQUENCE          02    Up to 5 fields and/or constants may be used to build the
Conditional Y/N? N      value of the result field.  For all except the last, you
Only if SYSFLAG =       must specify a value in the length column as follows:
                        Number = Start next component after that many characters
                          "&"  = Concatenate next component @ 1st trailing space
                          "+"  = Concatenate next component @ 2nd trailing space

   Result   F/C Field/Constant     Len      Result   F/C Field/Constant     Len
1           =                              3           =

2           =                              4           =
```

Figure 5-4. Set Field Equal to a Constant Operation Screen

This screen enables you to set a field equal to a value supplied by one or a combination of field(s) and/or constant(s). This permits you to construct up to four fields specified on the screen, @TSTFLDs, or @SYSBUF0, from up to five fields and/or constants.

You can specify up to five constants with a maximum length of 8 bytes each. For each component field or constant, you can specify a fixed length that causes the next component (if any) to begin at the next byte following the specified length. Alternately, specifying & in place of the field length concatenates the next component. This places the next component in the field starting at the first trailing space after the previous field, regardless of the field length. A + also specifies concatenation, but leaves one space between the two component fields.

A pass condition always results unless the specified components create a field length longer than the destination field. In this case, a fail condition results and the values are truncated to the length of the destination field.

If a field consists of numeric characters only, do not use this option to set that field equal to the value of a field that contains alphanumeric characters.

After you accept this screen, the Pass/Fail Action Specification screen (Figure 5-11) appears. Refer to Section 5.4.

5.3.3 Read A Record From A Data File

When you select FN 02 from the Edit Operation Type Selection screen (Figure 5-2), the Read a Record screen (Figure 5-5) appears.

```
IDEAS Screen Mask Editor          Read a record from a data file     Release 2.1

Screen      MAILSCRN      File access options
Field       CUST #        0  No Read, Set Field Only (No Optional Read Allwd)
SEQUENCE          33      1  Random record with key specified
Conditional Y/N? N        2  First logical record with key >= that specified
Only if SYSFLAG =         3  Next logical record

Data file                 Options 1 & 2 require a key specification below :
Access option =                   F/C Field/Constant                      Len
Only if SYSFLAG =         Key spec
Else use option
Error message on
Record read mode
Inquiry=I,Update=U

 If record is found set   Option 3 assumes the following :
 work buffer area for
 field/file:                 1. Option 2 Find First Logical record has been
 equal to record buffer         performed on the file prior to this read
 field/file:                 2. No sequential record read has been performed
 If record is found             on other files since last option 2 or 3 read.
 set System Flag =
```

Figure 5-5. Read a Record Screen

As indicated on the screen, you can make the entire record read and subsequent pass/fail logic conditional based on the value of a system flag.

After you specify whether the record read is conditional, enter the name of a data file. This screen allows you to specify that the program read a data record from any IDEAS data file that is open for the current application (and to which the runtime user has read access rights), even if the file is not associated with the screen you are defining.

The access options of the Read a Record Operation are:

0 -- NO RECORD READ, SET FIELD ONLY

This option, which is not actually a record read, allows the program to take a field from the most recently read record and use it to fill an appropriate field on the screen. This option is particularly useful for extracting multiple fields from an unassociated data file.

This option provides two functions. First, it allows you to use a field from the record buffer to supply a value to several fields in the work buffer. Second, it allows you to specify that several individual fields, but not the entire record, be read into the work buffer.

After you specify Option 0 in the box on the left side of the screen, the cursor immediately goes to the field at the bottom left of the screen that requests the name of the field in the work buffer that you wish to set equal to a field in the record buffer.

1 -- RANDOM RECORD WITH KEY SPECIFIED

Read a random record with the key specified on this screen.

2 -- FIRST LOGICAL RECORD WITH KEY GREATER THAN OR EQUAL TO THAT SPECIFIED

The key is specified on this screen.

3 -- NEXT LOGICAL RECORD

Read the next logical record. This option must be preceded with Option 2. You must not have performed any logical reads or saves on other files since the last Option 2, 3, or 4 read from this file.

4 -- NEXT LOGICAL RECORD AFTER DELETION IN SAME FILE

Read the next logical record after deletions in same file. You must perform deletions using Option 0 of the Pass/Fail Action Specification screen.

Options 2, 3, and 4 all involve sequential reads. You should limit these to one file at a time; in other words, complete a sequential read of one file (for example, FILE1) before performing a sequential read on another file (for example, FILE2). This is necessary because the sequential read of FILE2 destroys the sort array for the sequential read of FILE1. To circumvent this difficulty, you can specify that the key for the last FILE1 record read should be stored in a temporary field while the sequential read of FILE2 is performed. When FILE2's reads are completed, execute a record read to find the first logical record with a key greater than or equal to the key stored in the temporary field. This resets the read order back to its original point.

After you enter the access option, you can indicate whether the access option is conditional on a system flag and provide an alternate access option.

You can specify that errors encountered during record read be displayed for the user. This function does not affect any pass/fail messages that you specified.

If the file to be read is one of those associated with the current screen, the program can read the entire record for inquiry only, or for update/delete. This is also true if the file to be read is an alternate file for a file associated with a screen, even if the alternate file is not itself an associated file. However, if the file from which the program is reading a record is neither an associated file nor an alternate file for an associated file, the read can be performed only in an inquiry mode.

To save a record to the data file after you have updated it, use the Save a Record option of the Pass/Fail Action Specification screen, as described in Section 5.5.

CAUTION

> Use care when reading from a file that allows duplicate keys. If you specify the record read as inquiry only, but save that record back to the file, the record is saved as a duplicate record.

During a record read, the record is read into a special record buffer. A box in the lower left corner of the screen reads, "If record is found, set work buffer area for field/file: ######## equal to record buffer field/file: ########." Use this space to indicate whether you are reading the entire record from the record buffer into the screen work buffer or whether you are only reading a particular field into the work buffer. If you are reading from a file that is neither an associated file nor an alternate file for an associated file, you cannot read the entire record; you can only read a field from the record into a particular field on the screen or into an @TSTFLD.

When you specify that you are reading a record for update, the screen work buffer file is set to the record buffer file. This indicates that all fields in the record are read into their corresponding fields in the work buffer for the screen (the entire record is read). To specify that only a particular field from the record be read into a particular field in the work buffer, use FN 29 to backspace to these fields and override the defaults.

When you specify that you are reading a record for inquiry, no default appears in the work buffer area and record buffer area fields. You must specify the name of the particular field or file in the record that you are using.

As the screen specifies, the system can set a system flag to a requested value if it finds a valid record, and to the reverse value if it does not find a valid record. This occurs regardless of the protected or unprotected status of the record.

The following conditions determine pass conditions, fail conditions, and setting of the system flag:

On a record read in Inquiry mode:

If the record exists, a pass condition results. The system sets the system flag for record found and reads the record into the buffer, regardless of whether the record is protected.

If the record does not exist (or an end of file occurs with Option 2, 3, or 4), a fail condition results and the system does not set the system flag.

On a record read in Update/Delete Mode:

If the record exists and is unprotected, a pass condition results. The system sets the system flag for record found, and reads the record into the buffer.

If the record exists but is currently protected, a fail condition results. The system <u>does</u>, however, set the system flag for record found and reads the record into the buffer.

If the record does not exist or an end of file occurs, a fail condition results and the system does not set the system flag.

Read Options 2 and 3 require you to construct a key of up to five fields and/or constants. The file's key specifications determine the actual number of components and their lengths. A constant can be no longer than 28 bytes, regardless of the file's component key length specifications.

After you accept this screen, the Pass/Fail Action Specification screen (Figure 5-11) appears. Refer to Section 5.4.

Record Protection

When you read a record for update, it remains in a protected state until one of three things occurs:

1. You save a record to that file using the Save a Record Option of the Pass/Fail Action Specification screen.

2. You load the CANCEL module through the Load Module Option of the Pass/Fail Action Specification screen (you can also use this option to specify what the CANCEL module is)

3. You press FN 31 when the CANCEL key is enabled.

If you read a record and load a program that is not the CANCEL module before saving a record to the file (through the Load Module option of the Pass/Fail Action Specification screen), the record remains protected. Section 5.5 gives further information about the Read a Record Module and the Load Module.

You may sometimes find it necessary to maintain exclusive control of a data file throughout an extended processing sequence. Section 5.7 explains the procedure for doing this.

5.3.4 Perform Logical Tests

When you select FN 03 from the Edit Operation Type Selection screen (Figure 5-2), the Perform Logical Testing Operations screen (Figure 5-6) appears.

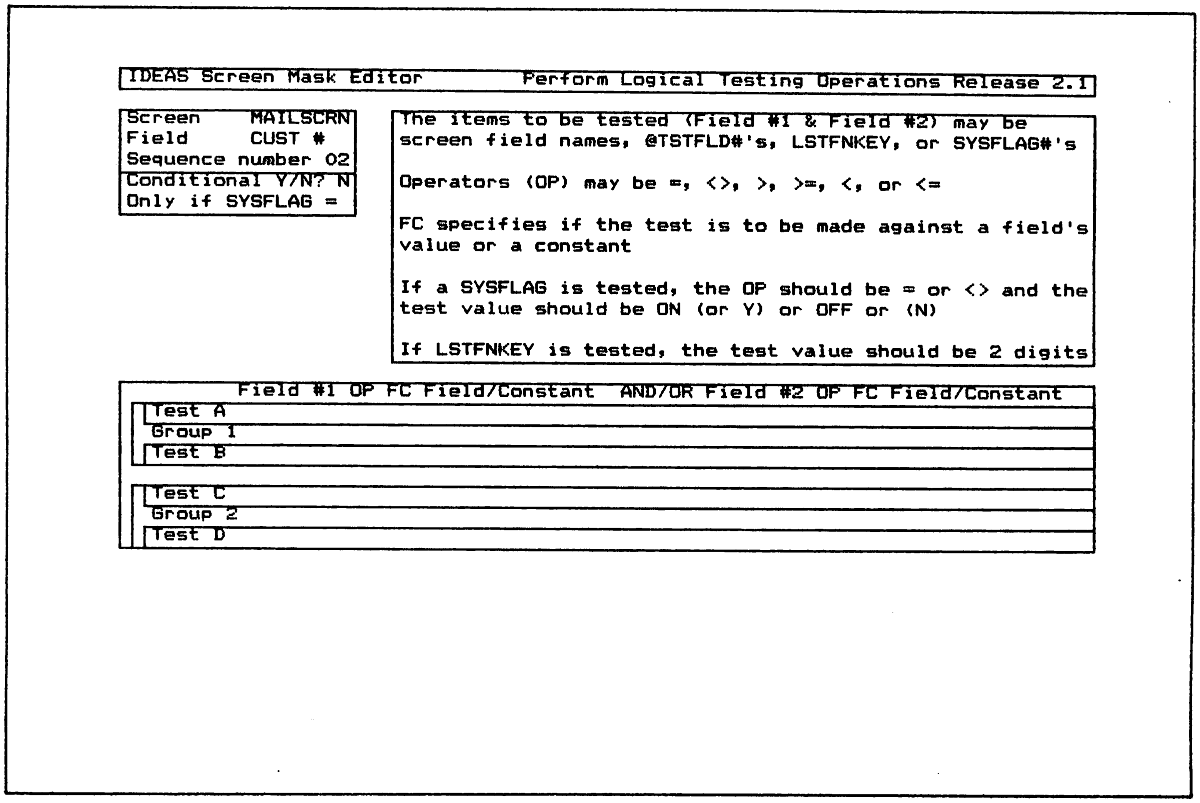


Figure 5-6. Perform Logical Testing Operations Screen

This option allows you to make up to eight logical comparisons with a wide variety of possible operands. Any specified combination of ANDs and/or ORs can constitute a pass action.

The possible comparisons are as follows:

Operand 1	Operators	Operand 2
Field or @TSTFLD	=,<,<=,>,>=,<>	Field, @TSTFLD, Constant
Field or @TSTFLD	=	BLANKFLD
SYSFLAG#	=	ON, OFF
LSTFNKEY (The value of the last function key pressed)	=,<,<=,>,>=,<>	Constant

Constants used can be up to 16 characters in length. The logic test performs the following procedure to obtain a single pass/fail result used to process subsequent logic:

1. The operation tests each field against a field or constant to obtain a single result.

2. Each pair of results undergoes the indicated Boolean operation (AND/OR) to obtain a single result.

3. Each pair of tests within a group undergoes the indicated Boolean operation (AND/OR) to obtain a single result.

4. The results from the two groups undergo the indicated Boolean operation (AND/OR) to obtain a single result.

After you accept this screen, the Pass/Fail Action Specification screen (Figure 5-11) appears. Refer to Section 5.4.

5.3.5 Math Calculations

When you select FN 04 from the Edit Operation Type Selection screen (Figure 5-2), the Math Calculations Specifications screen (Figure 5-7) appears.

```
IDEAS Screen Mask Editor        Math Calculation Specifications    Release 2.1

Screen      MAILSCRN       All calculations follow normal algebraic conventions
Field       CUST #         Result field may be screen field name or @TSTFLD#
Sequence number 02         Operands may be screen fields, @TSTFLD#, or CONSTNT#
Conditional  Y/N? N        OP1 may be blank, MAX, MIN, MOD, ABS, INT, or SGN
Only if SYSFLAG =          OP2 may be + , - , * , / , or ↑

Constants which may be used in operands below:    If OP1 is MAX or MIN, OP2-OP4
CONSTNT1 =              CONSTNT5 =                 are assumed to be commas. Any
CONSTNT2 =              CONSTNT6 =                 other OP codes are ignored.
CONSTNT3 =              CONSTNT7 =                 If OP1 is MOD, only Operands
CONSTNT4 =              CONSTNT8 =                 1 & 2 are used, comma assumed

   Result      OP1   Operand1  OP2  Operand2   OP3   Operand3   OP4  Operand4
1          =       (                         )     (                          )
2          =       (                         )     (                          )
3          =       (                         )     (                          )
4          =       (                         )     (                          )
5          =       (                         )     (                          )
6          =       (                         )     (                          )
7          =       (                         )     (                          )
8          =       (                         )     (                          )
```

Figure 5-7. Math Calculations Specifications Screen

Using this screen, you can set any combination of up to eight fields to values specified by math functions. Each math function can have up to four operands, including field values, @TSTFLDs, and constants. The math function can be an arithmetic operator (including add, subtract, multiply, divide, and raise to a power), or one of six BASIC-2 functions (MAX, MIN, MOD, ABS, INT, or SGN). A pass condition results unless there is a math error, such as division by zero or taking the square root of a negative number. Each set of math results can be conditional based on the value of a system flag.

After you accept this screen, the Pass/Fail Action Specification screen (Figure 5-11) appears. Refer to Section 5.4.

5.3.6 Range Test(s)

When you select FN 05 from the Edit Operation Type Selection screen (Figure 5-2), the Range Test Specification screen (Figure 5-8) appears.

```
IDEAS Screen Mask Editor        Range Test Specifications           Release 2.1

Screen     MAILSCRN       Up to 6 range tests may be specified.
Field      CUST #         Test values may be constants or field values.
Sequence number 02        FCN indicator meanings:
Conditional Y/N? N          F = Field name - test value is the field value
Only if SYSFLAG =           C = Constant   - test value is the specified constant
                                a blank is a valid constant
                                numeric fields require numeric constants
                            N = Not used   - the min or max value is not tested
                          ANY or ALL tests passing may imply a pass condition.

               '0 Any/All Pass condition exists if ANY tests pass

               FN Field name FCN Minimum value   FCN Maximum value
               '1
               '2
               '3
               '4
               '5
               '6
```

Figure 5-8. Range Test Specification Screen

This screen allows you to test up to six fields for range against 13-character constants or the contents of other fields. The six fields can be the same field (multiple ranges for one field), different fields, or some combination of the same and different fields. For each range, you can specify a minimum, a maximum (inclusive), or both. The fields and constants can be alphanumeric or numeric. You can specify that a pass condition results if any range check passes or if all of them pass.

After you accept this screen, the Pass/Fail Action Specification screen (Figure 5-11) appears. Refer to Section 5.4.

5.3.7 Table Look-Up Or Table Look-Up And Replace

When you select FN 06 from the Edit Operation Type Selection screen (Figure 5-2), the Table Look-Up and Replace screen (Figure 5-9) appears.

```
IDEAS Screen Mask Editor         Table look-up or Look-up & Replace Release 2.1

Screen      MAILSCRN   Test field name          Length      Max. mem. entries
Field       CUST #     Replacement to           Length      Max. mem. entries
Sequence number 02     Disk table file          Replace field from file
Conditional Y/N? N
Only if SYSFLAG =      If no test field, specify replace table only.  Item used
                       will be the one implied by the last memory table look-up.
                       For disk table, test field supplies the key to the file.
                       If memory table and disk table are both specified, disk
                       table is used only if there is no match in memory table.
Look-up table:


Replace table:

```

Figure 5-9. Table Look-Up and Replace Screen

This screen allows you to check the contents of a field against values listed in a table that are either in memory or in an IDEAS data file on disk. This operation is useful for converting flags, numeric codes, and dates to English equivalents for display purposes. A pass condition results if the value is in the table. IDEAS defines the maximum length of a table in memory as 240 characters, and limits the number of elements that can be in a memory table to INT(240/L) where L is the larger of the look-up field length or the replace field length, if any.

If you do not specify a test field, the system usea the one implied by the last memory table look-up. For the disk table, the test field supplies the key to the file. The field being tested does not have to be the field from which you specified the test. If you specify both memory table and disk table, the system uses the disk table only if there is no match in memory table.

If you specify a companion replace table, the system replaces the field in the look-up table with its corresponding element in the replace table. The replace field need not be the field you are table-testing -- it need not even be the same length. You can specify that both the table test field and the replace field are the same or less than their actual field length (except for the 1-byte function key field, which is considered 2 bytes long for table look-up). This gives you the ability, for example, to have two fields called "DATE" (six numeric) and "MONTH" (nine alphanumeric). Then you can replace the month part of the DATE field with the month spelled out in the MONTH field as follows:

```
Field to be tested = DATE
Len = 2 (although field length = 6)

Look-up table:
    "01","02","03","04","05","06","07","08","09","10","11","12"

Replace field  = MONTH     Len = 9

Replace table:
    "January  ","February ","March    ","April    ","May      ",
    "June     ","July     ","August   ","September","October  ",
    "November ","December "
```

For disk tables, the data file name is entered. For table look-up only, an IDEAS Type 1 file is best, and for look-up and replace, a Type 2 file is best. In either case, the key must be the table look-up value. For replace, you can use any additional field in the data file as a replace field.

After you accept this screen, the Pass/Fail Action Specification screen (Figure 5-11) appears. Refer to Section 5.4.

5.3.8 User-Supplied Process

When you select FN 07 from the Edit Operation Type Selection screen (Figure 5-2), the User-Supplied Process Specification screen (Figure 5-10) appears.

```
IDEAS Screen Mask Editor        User-coded Edit Specification        Release 2.1

Screen      MAILSCRN     This is the specification screen for a user-coded field
Field       CUST #       edit.  The actual user-coded program must be resident on
Sequence number 02       the same platter which contains the IDEAS utilities both
Conditional Y/N? N       at the time of specification and program generation.
Only if SYSFLAG =        Any values passed to or from the subroutine must be
                         specified as field names, @TSTFLD's, or @SYSBUFO.

 NOTE : DEFFN' Number must fall outside of the IDEAS reserved range ('32-'79)

Program name              (The name of the user-coded subroutine program module.)

Description

Call = DEFFN'
                          (                    ) is passed from field
                          (                    ) is passed from field
                          (                    ) is passed from field

                    Value (                    ) is returned to field
```

Figure 5-10. User-Supplied Process Specification Screen

A user-supplied process is a program module that the application designer writes according to certain specifications IDEAS requires. The IDEAS program generator integrates the process into the application program. The user-generated code passes an indicator that identifies the pass/fail condition back to IDEAS. As the screen indicates, the user-coded program must reside on the platter that contains the IDEAS utilities both during development and at runtime. Appendix C gives a more detailed discussion of this feature.

You should use this facility instead of modifying IDEAS-generated code manually. Modifying IDEAS-generated code might cause IDEAS to fail to recognize the code as an IDEAS program.

WARNING

Do not attempt to modify the IDEAS Release 2 code itself. Wang Laboratories, Inc., is not responsible for the performance of IDEAS utilities code that has been modified.

After you accept this screen, the Pass/Fail Action Specification screen (Figure 5-11) appears. Refer to Section 5.4.

5.4 PASS/FAIL OPTION SCREENS

The initial Pass/Fail Action Specification screen appears in Figure 5-11.

```
IDEAS Screen Mask Editor          Pass/Fail Action Specifications   Release 2.1

MAILSCRN CUST #    (02-0)     Touch EXECUTE to accept, or position cursor at the
                              desired P/F number using FN 5,6,12,&13 (or N,E,S,W)
  P=Pass, F=Fail, B=Both      then touch EDIT, DELETE (FN'9), or INSERT (FN'10)
    Action specification
      Operand1 Operand2
01     available              17     available              34     available
02     available              18     available              35     available
03     available              19     available              36     available
04     available              20     available              37     available
05     available              21     available              38     available
06     available              22     available              39     available
07     available              23     available              40     available
08     available              24     available              41     available
09     available              25     available              42     available
10     available              26     available              43     available
11     available              27     available              44     available
12     available              28     available              45     available
13     available              29     available              46     available
14     available              30     available              47     available
15     available              31     available              48     available
16     available              32     available              49     available
                              33     available              50     available
EXEC = Accept, EDIT = Modify, FN'0 = Op spec, '16 = Messages, '31 = Cancel
```

Figure 5-11. Pass/Fail Action Specification Screen 1

To add an operation, position the cursor key, then touch the EDIT key. The screen shown in Figure 5-12 appears.

```
IDEAS Screen Mask Editor          Pass/Fail Action Specification   Release 2.1

MAILSCRN  CUST #      (02-0)     Each action requires a condition P, F, or B and 1
                                 of the 26 pass/fail action specifications listed
     P=Pass, F=Fail, B=Both      below.  A blank may be used instead of an action
      Action Specification       specification as a no-op permitting any 2 of the
        Operand1 Operand2        additional tests and/or operations to be specified
No                               in operands 1 & 2.  Touch FN'15 at any time when
01 ■                             P/F specs permit to see possible tests/operations.

   A Enable skip ahead keys          N Receive indirect
   B Disable skip ahead keys         O Save record from operand 1
   C Enable skip back keys           P Save record indirect
   D Disable skip back keys          Q Branch to operand 1
   E Enable cancel key               R Branch indirect
   F Disable cancel key              S Load module in operand 1
   G Disable err msg/prompt          T Load module indirect
   H Display field/message/box       U Display operand 1-2
   I Copy operand 1 to operand 2     V Display indirect
   J Copy indirect                   W Change field attributes for operand 1
   K Transmit from operand 1         X Change field attributes indirect
   L Transmit indirect               Y Enable function key trap or "Help" screen
   M Receive to operand 1            Z Set or validate system or last used date
```

Figure 5-12. Pass/Fail Action Specification Screen 2

In general, you can select up to 50 actions on any individual pass/fail option screen. For each of the actions used, you must specify under what conditions the action is to occur by placing a letter in the pass/fail column on the specification screen:

P = Perform only if a pass condition exists
F = Perform only if a fail condition exists
B = Perform under both pass and fail conditions

Section 5.5 describes each pass/fail option in detail.

For each action used, you must enter an uppercase letter (A through Z) or a blank in the Opt column. This designates one of the 26 pass/fail options to occur if the condition is as specified in the pass/fail column. When editing a previously defined pass/fail, a blank in the pass/fail column erases that pass/fail specification. Entering a blank in the pass/fail column can also indicate additional conditions and operations, as described in Section 5.6.

5.4.1 Messages

You can define up to nine 64-character display messages for each pass/fail sequence. To define these messages, touch FN 16 from the first Pass/Fail Action Specification screen (Figure 5-11) and follow the instructions on that screen. You can use the same message as either an error message or a prompt; refer to Option H for instructions on displaying messages. Note that if a 64-character message is displayed on Row 24, execution of a field located at Row 24, Column 80 erases the 64th character of the message.

Option G allows you to disable the IDEAS system error messages that normally appear on Line 24. The option does not affect the messages displayed by means of Option H. A likely use for this option is for a record read in an add/change/delete situation: after you enter a key field, the system performs a record read. If the record is not on file, an error message appears, indicating that the record was not found. You need not add a record to the data file to see this message.

You may sometimes wish to disable an error message and then display it later. For an explanation of this option, refer to the discussion of DISPERRO under Pass/Fail Option H. Appendix H lists the IDEAS system error messages.

5.4.2 Advanced Pass/Fail Considerations

There are two eight-character fields next to the Option field, labeled OPERAND 1 and OPERAND 2. These generally specify the field names, file names, and screen names required to complete the action specification. When the specified operation does not require both fields, you can use the empty field to specify one of several additional operations and/or conditions. To free both operand fields for the additional operations, indicate a blank in the pass/fail option field. In this case, the previously established pass/fail condition determines the pass/fail condition.

Descriptions of the additional conditions and operations follow the descriptions of each pass/fail option in Section 5.6. You should note these conditions and operations before undertaking any pass/fail specification, since they include such important operations as skipping over pass/fail actions and setting system flags. As indicated on the screen, touching FN 15 from the Pass/Fail Action Specification Screen 2 (Figure 5-12) evokes a screen that summarizes some of the conditional and operational keywords.

Save Record (Options O and P), Transmit (Options K and L), Receive (Options M and N), and certain uses of Set Date (Option Z) reset the pass or fail indicator to reflect the success of the respective operations. Any subsequent pass/fail operations respond to this condition regardless of prior tests or operations performed. When you need to test these conditions after operations that change the pass/fail indicator, you can set systems flags to indicate pass or fail conditions, or you can use the extra conditions and operations described in Subsection 5.6.3.

5.4.3 Indirect Operations

Options J, L, N, P, R, T, V, and X are indirect operations. Basic data entry programs do not normally need indirect operations. Indirect operations can, however, prove useful for more involved applications. You may find it useful to familiarize yourself with standard pass/fail actions, as described in Section 5.5, before you attempt any indirect operations.

Indirect operations are operations that are indexed to variables rather than specified directly. This means that, instead of specifying the field to be processed directly, you specify a field whose contents specify the field to be processed. For example, assume there is a record with five different account fields, named 1 to 5. There is also a field on the screen labeled "account to be updated" (ACCTUPDT) that accepts a one-digit number from 1 to 5. The user keys an account number into field ACCTUPDT, and the cursor moves to that field to allow the user to update the current account name. To accomplish this, the field ACCTUPDT should have a post-entry edit, probably a range test that allows only the values 1 through 5. This edit should have an indirect branch as a pass/fail option (in this case, set to occur on a pass condition). (The indirect branch option is Pass/Fail Option R, with Operand 1 set to the field name ACCTUPDT).

In the above example, assume that the operator keys in a value of 3. The IDEAS program performs a range test and discovers that 3 is between 1 and 5; a pass condition exists. The program then performs the Pass/Fail actions set to occur on a pass condition. Specifically, it performs Option R and branches to a field. Unlike Option Q, however, it does not branch to the field whose name it finds in Operand 1 -- that is, field ACCTUPDT. In that case, the cursor would go back to the field the user just entered. Instead, the program looks in the field in Operand 1 -- field ACCTUPUDT -- to find the <u>name</u> of the field to which it should branch. In ACCTUPDT, it finds the value 3, so the program branches to the field whose name is 3.

As another example, assume there is a record with five different account fields, named 1 to 5. There is also a field on the screen labeled "account to be copied" (ACCTCOPY) that accepts a one-digit number from 1 to 5. Once an operator specifies a number in ACCTCOPY, an indirect branch to ACCTCOPY (Pass/Fail Option R) branches to the field specified in ACCTCOPY rather than ACCTCOPY itself. In other words, the following five fields exist:

Field name	Contents of Field
1	AGATHA
2	AMANDA
3	AMELIA
4	ALISHA
5	ALYSSA

If ACCTCOPY contains the number 4, an indirect branch that specifies ACCTCOPY branches to the field named 4, whose contents are ALISHA.

In this example, it would probably be more convenient if the five fields had names that were more alphanumeric, and were called ACCTNUM1, ACCTNUM2, etc. You can do this easily without requiring the operator to enter anything but the one-digit number ACCTCOPY and by using Operation 1, which is described in Section 5.3. Using this option, you can concatenate the one-digit field ACCTCOPY to the constant ACCTNUM, and you can place the results in a new field that you then reference in an indirect copy.

Some pass/fail options, such as branching operations, require you to specify only one operand. Other operations, such as copying the contents of one field to another, require two operands. An indirect operation that requires two operands indirectly references both operands. An example of this follows.

Assume the following fields:

Field	Length	Contents
WORKAMT	11	999999.99-
AMOUNT	11	blank
WKAMNAME	8	WORKAMT
AMTNAME	8	AMOUNT

An indirect copy that specifies that WKAMNAME be copied indirectly to AMTNAME copies the contents of the field specified in WKAMNAME to the contents of the field specified in AMTNAME. (The Pass/Fail Specification screen indicates this as B J WKAMNAME AMNTNAME) The results are:

WORKAMT	=	999999.99-
AMOUNT	=	999999.99-

The other fields remain unchanged.

Each of the "indirect" pass/fail operations listed has a direct correspondence to one of the direct options. You cannot specify added tests and operations indirectly; you can specify only standard pass/fail options indirectly.

This manual does not provide detailed descriptions of each indirect option, as they are fully explained under the referenced direct option. The only difference is that they operate indirectly, as explained above. You cannot combine direct and indirect operations in a single pass/fail option, except that you can use direct added tests and operations in indirect operations wherever you can use them in direct operations.

5.5 PASS/FAIL OPTIONS

The pass/fail options listed on Pass/Fail Action Specification screen 2 (Figure 5-12) are:

A -- ENABLE SKIP-AHEAD KEYS

Regardless of the screen-specified attribute, this enables the skip-ahead keys, allowing the operator to skip over fields (including fields defined as required fields), except for any fields that the operator cannot bypass. The skip-ahead keys are:

FN 04 - Skip to last field on the screen
FN 11 - Skip ahead five fields
FN 12 - Skip to the next field

The system processes skips before it processes any of the post-entry edits on the field where the operator touched the skip key. It performs any specified pre-entry edits on the field that is skipped to. Skips are determined in absolute field sequence so that even fields that cannot be displayed or entered from the keyboard are counted for skip-ahead or back. However, if the operator, using the skip-ahead key, lands on a field that cannot be entered from the keyboard, the cursor automatically skips ahead to the next field.

If fields that cannot be entered from the keyboard follow the last keyboard entry field, be sure to specify the last keyboard entry field as a field that the operator cannot bypass. If you do not do this, an enabled skip-ahead key moves the cursor to a field that the operator cannot enter from the keyboard. A field defined at Row 24, Column 80 defaults to a field that the operator cannot bypass.

B -- DISABLE SKIP-AHEAD KEYS

This prevents the operator from using the skip-ahead keys.

C -- ENABLE SKIP-BACK KEYS

This is the same as Option A, except it enables the skip-back keys. The skip-back keys are:

FN 07 - Skip to first field on the screen
FN 13 - Skip back one field
FN 14 - Skip back five fields

Refer to Option A for information on the processing of skip keys.

D -- DISABLE SKIP-BACK KEYS

This prevents the operator from using the skip-back keys.

E -- ENABLE CANCEL KEY

This permits the operator to use FN 31 to leave the current program automatically and return to the last menu encountered or to the value last entered as the CANCEL module using Pass/Fail Option S. With this option, IDEAS completes the CANCEL automatically when the operator presses FN 31, without requiring any additional Edit Specifications.

F -- DISABLE CANCEL KEY

This prevents the operator from using the CANCEL key (FN 31).

G -- DISABLE ERROR MESSAGES/PROMPT

This action disables the IDEAS system messages that appear on line 24 of the screen. It does not affect user-defined messages that appear there. It disables messages only until the system processes the next field (e.g., if you specify this option during pre-entry, it affects pre-entry to this field only. If you specify it during post-entry, it affects post-entry to this field and pre-entry to the next field).

Appendix H contains a list of the IDEAS system messages.

H -- DISPLAY FIELD, MESSAGE OR BOX

Sometimes it is necessary to display text that is not in a normal field position on the screen. This pass/fail option displays the contents of any screen field, @TSTFLD, display messages or error messages at any row and column position on the screen. It also allows you to display or erase a box. Generally the first operand indicates what to display and the second operand indicates where on the screen to display it.

To display a message defined for this pass/fail sequence, Operand 1 should be DISPMSG# or DISPERR#, where # is a number from 1 to 9. Both DISPMSG# and DISPERR# display the same nine messages defined by touching FN 16 on the first pass/fail screen. However, DISPERR# causes the audio alarm to sound when it displays the message, while DISMSG# does not. RINGBELL as Operand 1 causes the audio alarm to sound without displaying a message.

A special message command, DISPERR0, displays the last message encountered. DISPERR0 displays the IDEAS system error messages and the error messages you have defined by means of FN 16. If the last message encountered was an IDEAS system message, DISPERR0 displays it even if you evoked Option G (Disable Error Message). You can use this option when you read a record with the message disabled, but later wish to display the IDEAS system error message. For example, in an add/change/delete situation, you would disable the message so that an operator entering a new record would not see an error message indicating that a record cannot be found. However, if the record exists but is protected, the IDEAS system error message for this condition indicates which station is using that record. If you wish to display this to the operator, you should then use DISPERR0; there is no other way to display which station is using the record.

You should use DISPERR0 only when you have determined that an IDEAS system message would have appeared had the message not been disabled. Otherwise, an inappropriate message might be displayed. Typically, you should use this option as a pass/fail action after a record read/save.

If Operand 1 is a field name, @TESTFLD#, DISPMSG#, DISPERR#, or RINGBELL, Operand 2 can be a blank, an additional test or operation, or an eight-digit number of the form RRCCBBLL, where RR is the row, CC is the column, BB is the number of character spaces to erase before printing, and LL is the number of characters from the field to print. You must use all eight digits, and they must be in the proper positions. IDEAS counts from 1, not zero, so rows are from 1 to 24 and columns are from 1 to 80. If Operand 2 is blank, Row 24, Column 1 displays the item named in Operand 1.

If Operand 1 is PRINTBOX or ERASEBOX, Operand 2 must be an eight-digit number of the form RRCCDDWW, where RR is the row, CC is the column, DD is the depth, and WW is the width. Row and Column represent the upper left corner of the box you are defining. You can create lines by specifying a depth or width of zero.

Once a field is displayed anywhere other than on Line 24, it remains until the program erases it. To erase a message, fill a second field with blanks and display that field over the message.

I -- COPY OPERAND 1 TO OPERAND 2

This copies the contents of Operand 1 to Operand 2. This is an alphanumeric copy. If you use this option with numeric fields, you should exercise care to avoid truncation of those fields.

Copying to a record does not save the information on disk; you must use Pass/Fail Option 0 for that. Copying to a record replaces the entire record area of the work buffer with the information in Operand 1. By the same token, copying from a record is not a record read; use the Read a Record operation for that. Copying from a record copies the entire record position of the work buffer to the area specified by Operand 2.

You can use Option I to delete a record. To do this, copy BLANKFLD (all blanks) to the data file name or to File # X, where X is the file number of one of the associated data files specified on the Screen Mask Specification screen (Figure 4-3). Then save the data file using Pass/Fail Option 0.

You can use Option I to set a screen to blanks. To do this, copy BLANKFLD to the screen name and display the screen.

When copying from one field to another, be careful about copying a numeric to an alphanumeric field and vice versa. This could yield inaccurate results in later processing.

Operand 1 can be:

- A field name.

- @TSTFLDn (0 < = n < = 9).

- BLANKFLD (all blanks) -- Used to blank out an individual field or an entire screen.

- @SYSBUF0.

- PARTIT'N (two digits).

- TERMINAL -- The two-digit number that the BASIC-2 #TERM function returns.

- CPU ID # -- The five-digit number that the BASIC-2 #ID function returns.

- OPERATOR -- 3-byte user ID of current operator.

- USERCLAS -- 1-byte user class of current operator.

- NEWTRAN# -- Refer to the discussion of counters in Section 5.2.

- NEXTSEQ# -- Refer to the discussion of counters in Section 5.2.

- ERRORMSG -- The last IDEAS system message (refer to the discussion of messages in Subsection 5.2.14).

- @ERRMSG# -- The last system message number.

- An associated data file name -- The name of one of the data files specified on the Screen Mask Specification screen (Figure 4-3).

- FILE # n (1 < = n < = 7) -- The file number of one of the data files specified on the Screen Mask Specification screen.

- The screen name -- The current contents of all fields in the work buffer. Putting the screen name in Operand 1 copies the entire work buffer. You can use this operand to copy a file into the work buffer.

- HEX(hh) -- A valid hex code. If you specify a hex code as the field to be copied, the destination field completely fills with that hex code. This function is similar to the BASIC-2 ALL function.

- STATION# (two digits) -- Station number is unique to each partition on each CPU. If there is only one CPU, the station number is the partition number. If there is more than one CPU, the station number is the partition number plus a multiple of 16, depending on the number of CPUs.

Operand 2 can be:

- A field name
- An associated date file name
- @TSTFLDn (0 < = n < = 9)
- FILE # n (1 < = n < = 7)
- @SYSBUFO
- The screen name (work buffer)

J -- COPY INDIRECT

Indirect copy is the same as Option I, except that the field to be copied cannot be greater than 256 bytes in length.

Options K, L, M, and N involve telecommunications transmission and are described in Appendix G.

O -- SAVE RECORD FROM OPERAND 1

Operand 1 must contain the name of one of the seven data files associated with the current screen or the file number as defined in the Screen Mask Specification screen (Figure 4-3). The record associated with that data file is saved on disk. The Save Record option resets the pass/fail condition; if an error occurs (such as an illegal duplicate key), a fail condition ensues.

If you use this option after a read for update, it saves the current work buffer contents for that data file in the position the pointer for last record read indicates. Thus, if the runtime user changed the key for that record, a new record does not result. Instead, the record with the new key is saved in place of the old record. Whenever you perform a save without a read for update, the system automatically creates a new record and inserts it in the proper place, unless it is an illegal duplicate key.

To delete a record, you can copy a record of all blanks to the data file. Refer to Pass/Fail Option I for instructions.

Saving a record to a primary file automatically saves the record to any alternate files opened for the application.

Operand 2 can be an extra condition.

After saving a record, test for a fail condition.

P -- SAVE RECORD INDIRECT

Indirect save record is the same as Option O.

Q -- BRANCH TO OPERAND 1

Operand 1 contains the name of a field on the current screen. IDEAS branches to the specified field and begin execution of any pre-entry specifications for that field. Any subsequent pass/fail actions specified for the current field are ignored. If Operand 1 is the name of the current field, the program branches back and begins executing the first pre-entry edit for that field.

Operand 2 can be left blank, or can contain an added test or operation (this is the only operation that is processed prior to a branch).

If Operand 1 is not a field actually defined on a screen, a runtime error occurs because of an endless loop on the last-numbered field. IDEAS seeks to prevent this by not allowing invalid field names on direct branches at time of entry.

R -- BRANCH INDIRECT

Indirect branch is the same as Option Q.

S -- LOAD MODULE IN OPERAND 1

This option can perform one of two functions: it can redefine the module to load if you touch the CANCEL key (FN 31), or it can specify that IDEAS load another module at this point.

To load another module at this point, enter the name of the module to load in Operand 1. If it is "@CANCEL@", IDEAS loads the current cancel module. To load a screen, you must specify the name of the program that loads the screen. If the loaded screen is a subscreen without any fields, the attributes for the previous screen are retained.

If Operand 1 contains the name of a module to load, you can use Operand 2 to specify that certain variables be cleared when Operand 1 is loaded. If Operand 2 contains COMCLIDS, all IDEAS variables are cleared. If Operand 2 contains COMCLRPT, all IDEAS report variables are cleared. When chaining from a report to a screen, you should clear report variables.

To reset the name of the CANCEL module, enter the name of the new CANCEL module as Operand 1 and enter "ONCANCEL" as Operand 2. This does not load the new CANCEL module, it merely changes the specifications of the CANCEL module.

T -- LOAD MODULE INDIRECT

Indirect load is the same as Option R.

U -- DISPLAY OPERAND 1-2

This displays all of the displayable fields on the current screen, starting with the field named in Operand 1 and ending with the field named in Operand 2. Fields are always displayed in order of field number rather than physical position or buffer position.

If you leave Operand 1 blank, the display starts with the first field on the screen. If you leave Operand 2 blank, the display ends with the last field on the screen. You can have no other values in either of these fields.

Note that the Set Field Equal to Field and the Math Calculation edit operations (FN 01 and FN 04 on the Edit Operation Type Selection screen, Figure 5-2) automatically redisplay the fields in question. The Read a Record edit operation (FN 02 on Figure 5-2) does not automatically display any fields, and you must use Pass/Fail Option U if you wish to display the contents of the record.

V -- DISPLAY INDIRECT

Indirect display is the same as Option U.

W -- CHANGE FIELD ATTRIBUTES FOR OPERAND 1

This action allows you to change selected field attributes. Operand 1 must contain the name of a field on the screen. Operand 2 can be one of the following options. ROW and COL (column) indicate position on screen, while POS indicates position in the screen work buffer.

Set	Increment	Decrement	Display	Keyboard
ROW=nn	ROW+nn	ROW-nn	DSP=NONE	KBD=ON
COL=nn	COL+nn	COL-nn	DSP=NRML	KBD=OFF
POS=nnnn	POS+nnnn	POS-nnnn	DSP=BRGT	
			DSP=BLNK	

Operands 1 and 2 are required. No additional tests or operations are permitted.

No checks exist to ensure that you observe the boundaries of the screen (ROW 1 < = N < =24, COLUMN 1 < = N < =80) or the screen buffer while changing attributes. Be sure that POS function does not exceed buffer size or a system error stops program execution.

X -- CHANGE FIELD ATTRIBUTES INDIRECT

Change field attribute indirect is the same as Option W.

Y -- ENABLE FIELD LEVEL FUNCTION KEY TRAP OR HELP SCREEN

If Operand 1 contains HELPSCRN, Operand 2 must contain the name of the help screen you wish to enable or must be blank to disable a currently enabled help screen.

Operand 1 or both Operands 1 and 2 may contain a series of up to eight two-digit numbers that represent function keys that you can enter to this field and later process. You can also test these function keys as "last FN key" in later pass/fail actions. IDEAS always disables this function key trap after processing a field, so you must specify it prior to each field in which it is to be operable, either as pre-entry processing to the current field, or post-entry processing to the previous field.

Field-level function key traps enabled with Option Y override screen-level function key traps.

The two-digit numeric values of the non numeric function keys that you can use for this option (in addition to the numeric keys) are:

EDIT = 33   SHIFT/TAB = 35   FN(TAB) = 34

Operands 1 and 2 are required. No additional tests or operations are permitted.

Z -- SET OR VALIDATE SYSTEM OR LAST USED DATE

You can use this option to validate a date according to a defined format or to convert a date to a different format.

1. To validate a date, Operand 1 must contain a field name and Operand 2 must contain a format specification. IDEAS validates the contents of the field against the format specification and resets the current pass/fail condition according to the results of the validation. If you enable error messages, an error-message displays if validation fails.

   After validating a date, IDEAS stores it as the last used date. The date indicated when the current user entered the system, stored as the system date, serves as the last used date until you use Option Z to validate a date.

2. To convert the format of either the system date or the last used date (as defined in the preceding paragraph), Operand 1 must contain a format specification and Operand 2 must contain a field name. Note that this is exactly the reverse order as the first use of this option. IDEAS converts the indicated date to the appropriate format, stores it in the specified field, and sets a pass condition.

Format specifications (UPPERCASE = System, lowercase = last used):

MMDDYY MMMDDYY YYDDD
DDMMYY DDMMMYY DDDDD

In the above format specifications, MM yields a two-digit number between 01 and 12, while MMM yields a three-letter abbreviation for the month. DDDDD is a Julian date.

Operands 1 and 2 are required. No additional test or operations are permitted.

5.6 CONDITIONS AND OPERATIONS

In the cases where a pass/fail action specification only requires one specification or requires no field, you can use an empty specification field to specify one of several additional operations and/or conditions. You can indicate a blank in the pass/fail option field to free both operand fields for the additional operations; in this case, the previously established pass/fail condition determines the pass/fail condition.

5.6.1 Conditions

The use of IFFLG#ON and IFFLG#OF imposes an extra condition on the activity beyond that of the pass/fail condition. Regardless of whether the pass/fail condition is true, the specified system flag (SYSFLAG#) must be ON or OFF for the action to occur. A failure to meet these tests, however, does not change the state of a pass condition. The # in the test specification is the system flag designator and must be 0 through 9 or A through Z; for example, IFFLGAON tests flag A.

You can impose a similar condition based on the value of the last function key used (if any) since the last menu. Using a menu sets the last function key to the option specified from the menu. Both screen-level function keys and field-level function keys, described in Subsection 5.2.10, set the last function key value.

The use of a function key (FN 00 through FN 31), EDIT, EXECUTE, RETURN, FN(TAB), or SHIFT FN(TAB), as well as the 2236DW NORTH, SOUTH, EAST, WEST, INSERT, DELETE, NEXT SCREEN, PREVIOUS SCREEN, and SHIFT CANCEL turns on the "last function key used" flag.

The condition statements that test the value of the last function keys used are:

- IFLFK=ON -- tests for the function key flag ON state.
- IFLFK=OF -- tests for the function key flag OFF state.
- IFLFK=## -- tests for the value of the last function key.
- IFLFK<## -- tests if the value is less than that specified.
- IFLFK>## -- tests if the value is greater than that specified.

Table 5-1 lists the numeric values of the nonnumeric function keys you can use for additional conditions. Nonnumeric function keys of the 2236DW terminal that Table 5-1 does not list cannot specify an additional condition.

Table 5-1. Numeric Values of Nonnumeric Function Keys

Function Key	Value	2236DW Function Key	Value
EDIT	33	NORTH	06
EXECUTE	32	SOUTH	05
RETURN	32	EAST	12
FN	34	WEST	13
SHIFT FN	35	INSERT	10
		DELETE	09
		CANCEL	33
		TAB	34
		SHIFT TAB	35

In addition to testing function keys and system flags as added conditions in performing P/F operations, you can test the last IDEAS system message number by using IFERR###, where ### is the system error message number. Appendix H provides a list of IDEAS system messages and their associated numbers.

If both operands are free and you want two added conditions, you can use both Operand 1 and Operand 2 to specify a combination of the added conditions. For all of the added conditions, you can replace the first two letters (IF) with the letters OR. You can then combine the added conditions in the following ways:

- If the first two letters in the test are IF, the test specified in Operand 2 is ANDed with that specified in Operand 1 and the action occurs only if the pass/fail condition is true and <u>both</u> added tests are true.

- If there are two added tests, and the one in Operand 1 starts with IF and the one in Operand 2 starts with OR, the action occurs only if the pass/fail condition is true AND if <u>either</u> of the added tests is true.

- If there are two added tests, and the one in Operand 1 starts with OR and the one in Operand 2 starts with IF, the action occurs if the pass/fail condition is true, or if both of the added tests are true.

- If there are two added tests, and they both start with OR, the action occurs if the pass/fail condition is true, or if either of the added tests are true.

Table 5-2 shows some examples of additional conditions.

Table 5-2. Examples of Additional Conditions

P/F	Option	Operand 1	Operand 2	Action Performed
P	A	IFFLG7ON	IFLFK=31	Enables the skip-ahead keys if a PASS condition exists AND System Flag 7 is ON AND if the last function key used was FN 31.
P	A	IFFLG5ON	IFERR015	Enables the skip-ahead keys if a PASS condition exists AND System Flag 7 is ON AND if the last IDEAS system message number was 15.
F	B	IFFLG5OF	ORFLG6ON	Enables the skip-back keys if a FAIL condition exists AND if either System Flag 5 is OFF or System Flag 6 is ON.
P	A	ORLFK>00	IFLFK<15	Enables the skip-ahead keys if a PASS condition exists OR if the last function key used is greater than FN 00 and less than FN 15.
P	A	ORLFK=00	ORFLG0=ON	Enables the skip-ahead keys if a PASS condition exists OR if the last function key used was FN 00 OR if System Flag 0 is ON.

If only Operand 2 is free (when Operand 1 is required for the pass/fail action specification), you can add one condition in Operand 2 as either an AND (IF) or an OR (OR) situation.

5.6.2 Operations

When both Operand 1 and Operand 2 of the pass/fail specification are free, or when Operand 2 is free, you can specify certain operations to occur when the pass/fail condition is true. Table 5-3 describes these operations.

Table 5-3. Additional Operations

Operand	Action
FLAG#=ON	Sets the specified system flag to an ON state.
FLAG#=OF	Sets the specified system flag to an OFF state.
LSTFK=OF	Sets the "last used function key flag" to an OFF state.
LSTFK=##	Sets the "last used function key" flag to an ON state and sets an artificial value (##) for "the last function key used".
SEQ#=nnn	Sets the next default sequence number (NEXTSEQ#) to nnn.
SKIP##PF	Skips the next ## pass/fail activities. The number of pass/fail activities skipped or jumped back over is a relative number. Blank pass/fail specifications should not be counted and the number of pass/fails passed over should be equal to the number of active pass/fails <u>between</u>, but not including, the origin and the destination.
BACK##PF	Causes processing to jump back over ## pass/fails.

When both Operand 1 and Operand 2 are free, you can combine one operation and one added test. In this case, place the operation in Operand 1. Table 5-4 provides some examples of this.

Table 5-4. Examples of Conditions and Operations

P/F	Option	Operand 1	Operand 2	Action Performed
P	F	FLAGA=ON	IFLFK=31	Disables the CANCEL key and turns system flag A on if a pass condition exists and if the last function key used was FN 31.
F	E	SKIP03PF	ORFLG7OF	Enables the CANCEL key and skips the next three pass/fail operations if a Fail condition exists or if System Flag 7 is on.

5.6.3 Saving and Retrieving Pass/Fail Conditions

If a pass/fail action fails, you may want to branch back to the same action and try again. The failure of pass/fail Options O, P, K, L, M, N or Z, however, resets the pass/fail condition. In this case, you must recall the previous pass/fail condition or the previous error message. When this happens, you can do one of two things: you can set system flags to save a pass/fail condition, or you can use extra conditions and operations in the pass/fail actions immediately preceding and following the pass/fail action that changed the pass/fail condition and/or the error message. The extra conditions and operations are:

- SETOLDPF -- Saves an existing pass/fail condition prior to executing a pass/fail that will change the current state.

- GETOLDPF -- Recalls the pass/fail condition that was saved by means of SETOLDPF.

- IFOLDPFP -- Tests the pass/fail condition saved by means of SETOLDPF.

  IFOLDPFF
  OROLDPFP
  OROLDPFF

- IFERR### -- Tests for the last IDEAS system message number.
  ORERR### -- Appendix H lists the IDEAS system message numbers.

In addition to the above operations, you can also use the variable ERRORMSG or @ERRMSG# as Operand 1 in Pass/Fail Option I or J to copy the last IDEAS error message or error message number into any valid destination field.

5.7 SPECIAL TOPICS

There are some capabilities of traditional languages that you cannot specify directly through the pass/fail actions and logical edits listed above. You can perform most of these in other ways, however. For example, IDEAS provides no direct facility for using string functions in the pass/fail options, but you can do this by creating subfields and using them rather then trying to use a piece of a field. A user exit module that IDEAS can integrate into its generated application programs, as described in Appendix C, can perform other types of operations that IDEAS does not include, such as complicated formulas and check digit-calculations.

Exclusive File Access

You may sometimes find it necessary to maintain exclusive control of a file throughout an extended processing sequence. This procedure is sometimes known as locking a file. To lock a file, perform the following steps:

1. Establish a separate "access" data file that you will use solely to maintain a record of whether any data files are being accessed and whether they are being accessed exclusively. A record of this data file should contain a field with the name of the file being accessed, a field indicating whether access is general update or exclusive use, and a field identifying the station number accessing the file. The file should allow duplicate keys so that multiple users can update simultaneously.

2. To read for general (nonexclusive) update, perform the following steps:

   a. Before reading from the main data file, perform a read on the access file to check whether the main data file is in the access file and whether its use is exclusive. No read should be allowed on exclusive use.

   b. If the main file is not in the access file, write a record to the access file indicating the station number, the name of the file being read, and the fact that the read is for general (nonexclusive) update.

   c. At the completion of the nonexclusive read, delete the record from the access file.

3. To read for exclusive update, perform the following steps:

   a. Before reading, check the access file. If the main data file is on record, no exclusive update should be performed.

   b. If the main data file is not on record, write a record to the access file indicating the station number, the name of the file being read, and the fact that the read is for exclusive update.

   c. After all processing on the main data file is complete, delete the record from the access file.

Arrays

To simulate arrays with IDEAS, you can define a separate field for each element in the array and access the field with an indirect branch based on the components of the element. For example, to establish a one-dimensional, 10-element array using IDEAS, you can define ten fields called ARRAY001 to ARRAY010. When you wish to call a field, perform an indirect copy to a field that consists of the character string ARRAY plus the number of the element in the array. You can build the field name using the Set a Field option of the Edit Operation Type Selection screen (Figure 5-2).

To establish a two-dimensonal, 10-by-10 array, define 100 fields, called FLD.0101 to FLD.0110, FLD.0201 to FLD.0210, and so on to FLD.1001 to FLD.1010. When you wish to call a field, perform an indirect copy to a field that consists of the character string FLD. plus the first parameter of the array (the first two digits) plus the second parameter of the array (the next two digits).

End-Of-Period Rolldown

Many application screens display a series of related, sequential fields. When processing these fields, you may want to move the contents of each field to the subsequent field and update the first field. For example, if you have 12 fields that represent the transaction totals of the past 12 months, at the end of each month you move each field to the next field below and put the most recent transaction total in the first field.

You can use IDEAS to "rolldown" as described above without defining 11 or 12 different pass/fail actions. Instead, you can define two "superfields" in the same buffer positions as the month fields. The first superfield overlaps months 1 through 11, and the second superfield overlaps months 2 through 12. Copying the first superfield to the second superfield and then displaying the second superfield updates 11 months simultaneously. You can then copy blanks into the first month field and redisplay it. If you cannot fit a superfield into the 255-byte limit for fields that are not displayed, you can define more than one superfield and update the screen in sections.

Operator-Controlled Conditional Branches

You can define conditional branches based on operator selection by using an indirect branch based on the result of a Table Look-Up and Replace operation. To do this, define a separate Look-Up and Replace table for each field from which the operator can select an operation. Replace the field that the operator entered with the name of a workfield for which you define your desired operations. Branch indirectly to the replacement field.

CHAPTER 6
REPORTS

The IDEAS Release 2 Report utilities are conceptually similar to the Screen Definition utilities. After designating associated data files, you define a report mask and the operations to perform when printing the report. Like the screen mask, the report mask can include work fields, constants, @TSTFLDs, system defaults, and a variety of operations to produce a multilevel and complex report.

In its simplest form, an IDEAS report prints the contents of a data file according to a designated order and format. The Report utilities, of course, also provide the means for defining reports of a much greater complexity, which include information from several data files grouped according to a number of levels, a multilevel system of headers and footers, and an extensive amount of field processing.

6.1 OVERVIEW

There are five basic components of the Report Creation utility: File Reading, Line Formats and Fields, Operations, Level Breaks, and Record Selection. These components are summarized below. A more detailed screen-by-screen explanation of the Report utilities follows the overview.

The Report utilities require some special conventions in the use of System Flags 1 through G. Section 6.3 describes these conventions.

6.1.1 File Reading

The File Reading component of the report utility allows you to associate up to seven data files with the report. The first file specified establishes the order in which the records are read, since the data in file number one is read sequentially according to the logical key sequence of the file. You can specify that Files 2 through 7 be read automatically each time File 1 is read by defining key fields for these files. The file reading specifications can also indicate whether a file is to be read for inquiry or update. Section 6.3 describes file reading.

6.1.2 Line Formats and Fields

This component of the utility enables you to determine the appearance of the report by defining individual line formats. You define lines according to a hierarchy of line levels and a variety of line types. Line levels indicate the order in which the line formats print; lines of a particular level print until the value of a specified "level break" field changes. Line types include such distinctions as headers, footers, null lines, which do not print and are used for work fields, and test lines, which print separately from the rest of the report.

You specify report fields when you define line formats in the same way that you specify screen fields when you define screen masks: touch FN 00 and enter the appropriate field specifications in the screen that appears. Some of the report field attributes differ from screen field attributes. IDEAS provides four default report fields that indicate the current month, day, year, and page number of the report.

Refer to Section 6.5 for details on specifying line formats and fields.

6.1.3 Operations

Report operations allow you to manipulate or format data to print on a report. You specify report operations in the same manner as the Edit Specifications of the Screen utilities. Refer to Chapter 5 for information on Edit Specifications.

Each operation has an eight-character name and automatically receives a sequence number you can change later. You can indicate a level test and a system flag test to ensure that operations are performed on a selective basis.

Individual operations can include up to 64 Edit Specifications. The available Edit Specifications are the same as those for screens, except for those specified in Section 6.6, which presents further details of report operations.

6.1.4 Level Breaks

The lines of a report print according to a series of hierarchical levels, allowing for nesting of data amidst a series of leveled header lines and footer lines. Defined level breaks determine the levels within a report: a level break is a change in the value of a field that changes the current level of the line formats to print. You define the line levels themselves along with the line formats. The level break component of the Report utilities specifies the field that determines each level break, as well as whether a top of form follows the level break. Refer to Section 6.7 for information about defining level breaks.

6.1.5 Record Selection

The Record Selection component allows you to print only a portion of the data, according to range tests performed on key fields or logical tests performed on any fields read. The Record Selection component includes a sort specification screen that enables you to print the records of the report according to the logical sequence of up to five fields of the report. You can make record selection operation modifiable at runtime, allowing the operator to specify what sections to print out and in what order. Refer to Section 6.9 for information on record selection.

6.2 REPORT GENERATOR: MAIN SPECIFICATIONS

After choosing the Create option from the Report Utilities Menu, specify the report name. The Main Report Specification screen (Figure 6-1) appears.

```
IDEAS Report Generator - Main Specification Screen                Release 2.1

Module name      MAILREPT                              Revision number
Description      ################################      Last revised 09/27/83 SJL

Version               Edit & Document Privilege       Execution Privilege
Application           User class      User ID         User class      User ID
Function              Password                        Password

        Data files to be used in the module           Current record selection:
FN      File        Description                       Range    User modifiable
01                                                    Logic    User modifiable
02                                                    Sort     User modifiable
03                                                      by:
04
05
06
07

08  Save program REMs    N   Number of files     0    Number of fields         0
09  No. of lines/page   55   Work buffer      1750    Number of line formats   0
10  Program protection   N   Record buffer     256    Number of operations     0
```

Figure 6-1. Main Report Specification Screen

The cursor goes to the first nine fields in the order they appear on the screen (Description, Version, Application, Function, Edit and Document Privilege, Execution Privilege). These fields contain the same information as the fields of the same names in the Screen and Data File Creation utilities. Refer to Section 3.2.

On this screen, you can associate up to seven data files with the report. These files can be primary or alternate key files. The file entered as File 1 is the sequence file that supplies the regular sequence for the report according to existing logical key sequence in that file. The Sort Specification Screen of the Record Selection Component (Figure 6-11) can determine alternate orders of printing records. For information on Sort Specifications, refer to Section 6.9.

To read files other than the sequence file, which is read according to its logical key sequence, you must specify a random access key for those files. Do this when the Report Generator File Reading Specification screen appears (Figure 6-2), described in Section 6.3. All files for which you specify keys on this screen are read automatically each time a record in the sequence file is read.

You need not specify reference files (files from which single fields or records are read on an intermittent basis) as associated data files. Because the report execution modules read all seven associated data files once per cycle, to include files to be read only at the start or completion of the report increases the execution time and memory requirements needlessly. Reference files are read using the Read a Record operation, as described in Section 6.6.

A reference file that is not an associated file can read a record only. You should always specify files to which records are written within the seven associated data files.

The Main Specification screen of the Report Generator also contains three user-modifiable options, accessed through FN 08, 09, and 10.

FN 08 -- SAVE PROGRAM REMS
The report utilities create a control file and a report program. A Y indicates that the report program will include internal comments.

FN 09 -- NO. OF LINES/PAGE
The number of lines to be printed on a report before a top of form is issued. This defaults to 55. Section 6.8 explains the effect this field has on page breaks.

FN 10 -- PROGRAM PROTECTION
A Y indicates that the created report program will be program-protected, modifiable only through the IDEAS utilities.

The Main Specification screen displays any record selection criteria previously defined for the screen as well as whether these criteria are user-modifiable. If a sort was specified previously, the screen displays the file that contains the sort key field. Additionally, the Main Specification screen displays the number of files, fields, line formats, and operations previously defined for the report. The screen also displays the size of the record buffer and work buffer needed for the report.

6.3 FILE READING SPECIFICATIONS

When you first define a report, or if you revise a report and modify the associated files on this screen, accepting the Main Report Specification screen causes the File Reading Key Specification screen (Figure 6-2) to appear. When you are revising a report, accepting this screen causes the Report Utility Selection screen (Figure 6-3) to appear, as described in Section 6.4.

```
IDEAS Report Generator - Additional Data File Key Specifications   Release 2.1

Report program name is MAILREPT  Description: PRINTOUT OF MAILING LIST

#   File     Associated data file description Comment
1 MAILLIST CUSTOMERS AND ADDRESSES            File is read automatically
2                                             Associated data file 2 is not used
3                                             Associated data file 3 is not used
4                                             Associated data file 4 is not used
5                                             Associated data file 5 is not used
6                                             Associated data file 6 is not used
7                                             Associated data file 7 is not used
For each file used (2-7) specify the construction of the  random access key by
using a field name and file no. to identify a field in a lower-numbered file.
Specify under "I/U" if the file is to read in inquiry (I) or update (U) mode.
FN   File   I/U #fields Field/File Field/File Field/File Field/File Field/File
'1 MAILLIST  I     File is read automatically
'2                          /          /          /          /          /
'3                          /          /          /          /          /
'4                          /          /          /          /          /
'5                          /          /          /          /          /
'6                          /          /          /          /          /
'7                          /          /          /          /          /
EXECUTE = accept,  FN'0 = field names,  FN'1-'7= file specs,  FN'31 = cancel
```

Figure 6-2. File Reading Key Specification Screen

Unless you use any of the sorting options described in Section 6.9, IDEAS reads and prints File 1 according to the logical sequence of its key field. The system reads Files 2 through 7 randomly, according to the key you specify on this screen. The key can contain up to five fields. To specify the key fields, touch the appropriate function key and fill in the required information.

You must construct the key for any given file except File 1 from the lower-numbered files. That is, the record for File 1 must contain all of the fields necessary to construct the random key for File 2. Files 1 and 2 must contain all of the fields necessary to construct the key for File 3, and so forth.

IDEAS can read all associated data files, including File 1, either for inquiry only or for update. This allows you to create a report that both reports on and updates a file. For example, you might want to process a transaction file, print records of the transactions, update the master records, and delete the transactions. When reading any file for update, you must use a pass/fail action to save a record if the record should be rewritten. Otherwise, the record remains protected.

When you specify a key for an associated data file, and that file is read for update, the report automatically reads that data file at each processing (that is, each time a record is read from File 1).

If you do not specify a key for an associated data file, the report only reads the data file when you specify a Read a Record operation (described in Section 6.6), even though space is always allocated in the record buffer. In these cases, the record is read according to the key you specified when you defined the read operation. This means that if a record is read for update from an associated data file that has no key construction specified, the record remains the same in the buffer until you perform another record read on the data file.

If an associated data file with a specified key is read for inquiry, the report does not read it automatically each time it reads File 1; instead, the report reads it only when the value of its key field changes.

Associated data files that the report reads automatically by means of the key fields provided on this screen follow some specific conventions in the use of system flags. The report generator follows these conventions automatically. The report generator does not follow these conventions for data files that are not specified on the File Reading Key Specification screen (Figure 6-2). You can use the Read a Record operation, described in Section 6.6, to read such files, but you must maintain any flag conventions within the operation specification. System flag use conventions include:

1. SYSFLAG1 is always ON until the end-of-file of File 1 is reached. You should specify that any report level operations that should occur only at end-of-file should occur conditionally with Flag 1 OFF. Section 6.6 gives instructions for defining report-level operations.

2. SYSFLAG2 to SYSFLAG7 correspond to one of the associated data files (Data Files 2 through 7). Each of these flags, if used, is set to ON if the record is found, regardless of whether it is currently protected (in use by another partition); the flag is set to OFF if the record is not found.

3. SYSFLAGA to SYSFLAGG correspond to Data Files 1 through 7 only if the associated data file is read for update. If a record is found and is available (not protected by another user), the flag is set to ON. If the record is not found or is found protected, the flag is set to OFF.

Once the system flags are set, they can be tested in the various components of the Report utilities.

As indicated on the bottom of the screen, touching FN 00 provides a list of the field names for Files 1 through 7, as well as a list of the fields defined specifically for the report.

After you accept the File Reading Key Specification screen, the Report Utility Selection screen (Figure 6-3) appears.

WARNING

> If you change which data files are associated with the file numbers (for example, if you switch File 2 with File 3) on the File Reading Key Specification screen after you have defined the remainder of the components for the report, you must redefine the report components. Failure to do this causes incorrect file referencing during report execution.

6.4 REPORT UTILITY FUNCTION SELECTION

```
IDEAS Report Generator - Summary & Function Selection Module        Release 2.1

Report program name is MAILREPT
Description PRINTOUT OF MAILING LIST

      File    Description                         ---- Summary ----
1  MAILLIST CUSTOMERS AND ADDRESSES            Number of fields            0
2                                                      alpha               0
3                                                      numeric             0
4                                              Number of operations        0
5                                              Number of line formats      0
6                                              Maximum line length         0
7                                              Number of level breaks      0

FN Specification Options      Used  Modifiable   FN Specification Module    Used
'0 Field descriptions:         No   @ run time   -- --------------------    ----
   Record selection (Range)    No       No       '5 Line formats & fields    No
   Record selection (Logic)    No       No       '6 Operations               No
   Sort order specification    No       No       '7 File reading specs       Yes
'4 Default printer address     SYS      No       '8 Level break specs        No

(If default printer address is "SYS", address set in the START module is used)

Attention:      Touch FN'0-'8 listed above, EXECUTE to accept, FN'31 to cancel ■
```

Figure 6-3. Report Utility Selection Screen

This screen presents a summary of the report you are defining and a listing of the files it uses. The box in the lower left quarter of the screen indicates the record selection component of the utilities. Touching FN 00 evokes all record selection screens, as described in Section 6.9. If you defined any record selection criteria previously, FN 01, FN 02, and FN 03 allow access to Range, Logic, and Sort specifications, respectively. You can modify each specification screen separately. Accepting a specification screen returns you to the Report Utility Selection screen (Figure 6-3).

Touching FN 04 allows you to specify a default printer address for the operator. The printer address may be a valid printer (204 or 215 through 219), the CRT screen (005), or the printer listed in the last START module encountered ("SYS"). A report printed to the screen displays one screenful at a time, with all "before" and "after" line skipping specifications ignored (blank lines are printed). You can also specify that this value be modifiable at execution. If you do so, the specified value is the default value supplied at runtime when the system requests the printer address.

The box in the lower right quarter of the screen contains a menu of the remaining four components of the report generator and an indication of whether they have been used. FN 07 returns you to the File Reading Key Specification screen, described in Section 6.3. Sections 6.5 through 6.9 describe the components indicated by FN 00, FN 05, FN 06, and FN 08.

6.5 LINE FORMATS AND FIELDS

Touching FN 05 from the Report Utility Selection screen (Figure 6-3) causes the Report Mask Format Line Specification screen (Figure 6-4) to appear.

```
IDEAS Report Generator - Report Mask Format Line Specifications    Release 2.1

Print control  Line Format Image Window          Cursor is at line  1, column   1
No T L FS B A  Window below currently displays lines  1 to 12, columns  1 to  65
01
02
03
04
05
06
07
08
09
10
11
12
               1↑                                                            65↑
T = Line type    L = Level    FS = SYSFLAG# & state for  B = Extra line feeds
  B=Banner         Blank        optional sysflag test      before line is printed
  D=Detail         0-9          If used, line prints     A = Extra line feeds
  F=Footer         P = Page     only if sysflag state      after line is printed,
  H=Header         R = Report   AND level are correct.     in addition to the 1
  T=Test  N=Null                State: "Y"=ON, "N"=OFF     issued automatically.
'0=Define/edit field, '16=Edit print control specs,  EXECUTE=accept
```

Figure 6-4. Report Mask Format Line Specification Screen

Specifying text and fields for a report mask is similar to specifying text and fields on a screen mask in that you enter text by typing it directly onto the mask and you define fields by positioning the cursor and touching FN 00. Report masks, however, specify individual line formats that are repeated, in an order determined by line level and line type, for each record of the data file.

The Report Mask Format Line Specification screen appears with the cursor in the Line Format Image Window, where you specify text and fields. The following function keys work just as they do when you are defining a screen mask, as described in Subsection 4.1.2: FN 00 (define a field), FN 05 (cursor down), FN 06 (cursor up), FN 11 (cursor five spaces right), FN 12 (cursor right), FN 13 (cursor left), FN 14 (cursor five spaces left), FN 21 (move text and fields down), and FN 22 (move text and fields up). FN 09 and FN 10 operate on a report mask the way that FN 25 and FN 26 operate on a screen mask. FN 25 and FN 26 have no effect on a report mask.

You can specify up to 99 formats on the report mask. The maximum line length is 158 columns. A window on the screen displays 12 lines and 65 columns of the mask at a time. The current line number and column number of the cursor position and the boundaries of the window are displayed continuously. The window provides automatic horizontal scrolling in three steps, displaying columns 1 through 65, 47 through 111, or 94 through 158. There is an overlap of almost 20 columns.

You cannot cancel out of the Report Mask Format Line Specification screen. To leave this screen, you must accept it to return to the Report Utility Selection screen (Figure 6-3).

6.5.1 Print Control

Print control determines such things as report and page headers and footers, levels of report lines, and conditional tests for printing individual lines. Each format line of the report mask has a line type and level. In its simplest form, a report consists of a repeating detail line nested among up to 10 levels of header and footer lines, which in themselves you can nest among page and report headers and footers. A description of the line types and levels and an explanation of the printing order follow. A line may be of type T, B, H, D, F, or N.

T -- TEST

A test line is a line printed before you execute the report program. It enables an operator to line up a printer, or to check for proper paper or forms. If you specify one or more test lines, the following occurs before the report prints:

1. Execution halts. A prompt appears on the terminal and instructs the operator to check for proper paper and forms.

2. When the operator touches EXECUTE, the printer issues a top-of-form and all test lines print.

3. Execution halts again and a prompt appears on the terminal that gives the operator the choice of printing the test line(s) again or of printing the report. If the operator chooses to print the report, the printer issues a top-of-form and the report begins.

B -- BANNER

If you specify one or more banner lines, a banner page followed by a top-of-form prints prior to the main report.

A banner page consists of boxed information that includes the name of the report and the ID of the operator printing the report, a listing of all record selection and sort options used, and any banner lines specified on the Format Line Specification screen. A banner line can be blank.

H -- HEADER

You can specify header lines for report level, page level, and any grouping level (0 through 9). Header lines can contain text and/or fields.

D -- DETAIL

A detail line is the innermost level of line nesting, and has no level (the level parameter is left blank). A detail line can contain text and/or fields.

F -- FOOTER
Like header lines, you can specify footer lines for report level, page level, and any grouping level (0 through 9). Footer lines can contain text and/or fields. When defining fields in a footer, note the special considerations described in Subsection 6.5.4.

N -- NULL
Null lines, which do not print, serve the same function as fields on the screen mask that are not displayed. You can establish fields on null lines to use as work fields, or you can use the null lines as a form of internal documentation for the report mask. You do not need to specify a line level.

The line levels supported are:

-- BLANK
A blank level is used for detail lines, banner lines, null lines, and test lines. Specify a blank line level by leaving the line level field (Field L) blank.

0 through 9
These are hierarchical levels where 0 is closest to the detail line.

P -- PAGE
Header and footer lines specified as level P print at the top and bottom of each page of the report.

R -- REPORT
Header and footer lines specified as level R print only at the beginning and end of a report.

6.5.2 Printing Order

The printing order of an IDEAS Release 2 report is as follows:

1. Following the execution of test lines and the printing of any banner lines, the header lines print in the following order: P, R, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.

2. Detail lines print until a level break occurs. A level break occurs whenever the value of a field specified as a level break field in one of the associated data files changes. You can define level break fields by choosing the Level Break Specification component from the Report Utility Selection screen (Figure 6-3). Specify a field name and a file for each desired level. A particular level break can cause the printer to issue a top-of-form after all of the footer lines up to that level have printed. Section 6.7 presents the Level Break Specification screen.

3. Footer lines print for all levels from 0 to the level specified as the level of the previously encountered level break.

4. Header lines print in descending order, starting at the level at which the break occurred.

5. Detail lines print until another level break occurs.

6. At the conclusion of the data, all footer lines print in ascending order, followed by the report footer, and finally the page footer.

Page header lines and page footer lines print on the first and last lines, respectively, of every page, regardless of the current level.

As the screen indicates, you can make line printing conditional to the state of a system flag. You can also specify up to 9 blank lines to print before and/or after each defined format line.

For a complete description of processing order in a report, including operations as well as printing order, refer to Section 6.8.

6.5.3 Report Fields

You can specify up to 249 fields for the report. The fields can be part of the seven associated data files, or they can be unique to the report. Field length should not exceed the line length of the report.

To define a field, touch FN 00 when defining the line format in which the field should appear. This evokes the Report Field Specification screen (Figure 6-5), which enables you to define a field at the current cursor position.

```
IDEAS Report Generator - Field Specifications                         Release 2.1

Report program name is: MAILREPT    #   File    Description
PRINTOUT OF MAILING LIST            1 MAILLIST CUSTOMERS AND ADDRESSES
                                    2
Special (reserved) Field Names:     3
"@SYSPAGE" = Current page number    4
"@SYSMNTH" = System date month      5
"@SYSDAY " = System date day        6
"@SYSYEAR" = System date year       7

FN Attribute                FN Attribute                FN Attribute
00 File number              06 Field type           2   11 Sign code
       name                    0 = unsigned numeric        0 = "-" before
01 Field name  FIELD001        1 = signed numeric          1 = "-" after
02 Line               2        2 = alphanumeric            2 = "DB" after (-)
03 Column            49     07 Justify              N      3 = "CR" after (-)
04 Length (memory)    0     08 Zero fill            N   12 Commas             N
          (report)    0     09 Decimals             0   13 Floating "$"       N
05 Position          80     10 Grouping level           14 Blank after print  N

Note: Special date field names yield 2-digit values.  Page number is 4 digits.

Attribute FNs shown above, EXEC=Accept, 16=Field names, 25=Delete, 31=Cancel
```

Figure 6-5. Report Field Specification Screen

The following fields appear on the screen:

FN 00 -- FILE NUMBER

If the field you are defining is a field from one of the associated data files, enter the number of the data file. The name of the data file then appears below the file number. If the field is not part of an associated data file, enter 0 as the file number.

FN 01 -- FIELD NAME

If the field is from an associated data file, many of the remaining fields on this screen default to the parameters you entered during data file creation.

IDEAS provides four default field names that provide the current page number, month, day, and year. The field names and the values these fields supply are:

Field Name	Value Supplied
@SYSPAGE	A 4-byte numeric field containing the current page number. It is right-justified and zero-suppressed.
@SYSMNTH	A 2-byte numeric field containing the month digits of the system date. It is right-justified and zero-filled.

@SYSDAY — A 2-byte numeric field containing the day digits of the system date. It is right-justified and zero-filled.

@SYSYEAR — A 2-byte numeric field containing the year digits of the system date. It is right-justified and zero-filled.

The Table Look-Up and Replace operation makes it easy to convert the values returned in the default fields to any format.

FN 02 -- LINE
FN 03 -- COLUMN
The line and column of the field default to the line and column in which you touched FN 00. You cannot change the field position.

FN 04 -- LENGTH
Two length fields are shown: memory length and report length.

Memory length represents the length of the field within the data record of the work buffer. This is fixed if the field is from an associated data file. You must specify the memory length for a new field in the report. Memory length must include any sign and decimal point. For example, if the field might contain the number -99.99, the length must be at least six characters to accommodate. You cannot define a field that will overlap an existing field or existing text, and you must leave at least one character that is not part of either field between defined fields.

You cannot modify the report length. It is computed automatically and indicates the actual number of character positions that the field will occupy on the output report line. This differs from memory length if the field is a numeric field and you use FN 11, 12, or 13 (as described below) to indicate that the output should include a sign type, a dollar sign, or commas.

FN 05 -- POSITION
This is the position of the field within the work buffer. When the field is associated with a file, the position defaults to the starting position within the work buffer of the data file itself plus the relative position of the field within the file.

FN 06 -- FIELD TYPE
There are only three field types supported in the report generator, as opposed to 10 in the screen generator; they are:

0 Unsigned numerics
1 Signed numerics
2 Alphanumerics

Only Type 1 fields contain signs. FN 11 determines the manner in which the sign prints, as described below.

When a field is extracted from an associated data file, field type defaults to a type compatible with that defined for the data file. You can change this when defining a report. Due to conventions used to speed internal processing of calculation in reports, you should change numeric fields to alphanumeric fields in a report unless you will use them specifically to perform math calculations.

> CAUTION
>
> Use extreme care when changing the attributes of a field of a record that is read for update so that the integrity of the data remains intact.

All numeric fields in a report print as right-justified, space-filled. This makes it particularly important to change a numeric field to an alphanumeric when you want to print the field with its leading zeros intact (for example, a zip code). If you will use a field in math calculations <u>and</u> print it as an alpha-field, you must specify the field twice on the report mask: once as an alphanumeric field in a line format that will print, and once as a numeric work field on a null line (a line that will not print).

FN 07 -- JUSTIFY

A Y response in this attribute right-justifies the field display on the screen regardless of input size or type. An N response implies no right-justification.

FN 08 -- ZERO FILL

Refer to the discussion of FN 06, field type, for information on the manner in which numeric fields print.

FN 09 -- DECIMALS

A numeric response indicates the number of decimal places numeric fields should display.

FN 10 -- GROUPING LEVEL

The grouping level option is similar to the level break option, except that it specifies level breaks at the field level instead of at the line level. A field with a specified grouping level prints only after a level break of the specified level or a page break occurs. (Section 6.8 gives more information on what prints when a page break occurs.) Use the grouping level option when there are multiple fields on a line, but you wish to print only one field at a certain line level. This allows you to enhance the appearance of a report by removing extraneous clutter or to print out totals and subtotals without adding additional line formats.

An example of using grouping level is a report showing quarterly sales by salesman:

```
Salesman #1     1st qtr    ###,###.##
Salesman #1     2nd qtr    ###,###.##
Salesman #1     3rd qtr    ###,###.##
Salesman #1     4th qtr    ###,###.##
                           Total     ###,###.##

Salesman #2     1st qtr    ###,###.##
Salesman #2     2nd qtr    ###,###.##   etc.
```

If you group the items by salesman and the salesman's name prints only once for each group of records, there is less clutter:

```
Salesman # 1    1st qtr    ###,###.##
                2nd qtr    ###,###.##
                3rd qtr    ###,###.##
                4th qtr    ###,###.##
                Total      ###,###.##

Salesman # 2    1st qtr    ###,###.##
                2nd qtr    ###,###.##   etc.
```

You can do this by having two different line specifications at the detail level and setting and testing a system flag for salesman's name. An easier way is to note that there is obviously a level break on salesman (by name, number, etc.). Assume that this is a Level 0 break. The detail line is level " " (blank), but you can assign a grouping level of 0 to the salesman's name field. This means that it will print only on the first detail line following a level break of 0 or greater. The field specified as a grouping level does not have to be the field causing the level break, although it could be; in this case the field is the saleman's name, but the level break might be salesman number.

Another use of grouping levels is shown in the following example:

```
AREA        DISTRICT         SALESMEN     CUSTOMER
Western     Southern Cal     John Doe     Rockwell
                                          Hughes
                             Jim Smith    ABC, INC.
            Northern Cal     Mary Jones   Bank of Amer.
Central     Chicago          Sam Green    ZYX, Ltd.
```

In this case, the column headings might be a page header. A single detail line with four fields, each having a different grouping level, represents all printed lines.

1. The customer name has no grouping level (level is blank), so it always prints.

2. The salesman's name has a Level 0 grouping level; a Level 0 break is specified when the salesman name changes.

3. The district has a Level 1 grouping level; a Level 1 break is specified when the district changes.

4. The area has a Level 2 grouping level; a Level 2 break is specified when the area changes.

As mentioned above, a page break causes all grouping levels to print.

FN 11 -- SIGN CODE

Only Type 1 fields can contain signs. The sign prints only if the value contained in the field is negative. There are four formats for printing the sign:

Value	Sign code	Printed
-1234	0 = - before	-1234
"	1 = - after	1234-
"	2 = DB after (-)	1234(DB)
"	3 = CR after (-)	1234(CR)

CR and DB are considered signs. Therefore, they are displayed only for negative values.

FN 12 -- COMMAS

A Y response edits commas into the printed numeric value.

FN 13 -- FLOATING "$"

A Y response prints a dollar sign before the highest-order digit. If the field is a negative number, the dollar sign prints between the negative sign and the highest-order digit. For example, if you use both Options 12 and 13 and the sign code is 0, 1234567.89 prints as $1,234,567.89 and -123456.78 prints as -$123,456.78. If the sign code is 3, then -123456.78 prints as $123,456.78CR.

FN 14 -- BLANK AFTER PRINT

When computing totals, you must tell the report if it should blank out a file after it prints. For example, you should reset the salesman total field in the grouping level example above to zero after you process the footers for each Level 0 break. You can specify this with a Y response to this option. You do not need this option on fields that are read as parts of data records or computed for detail lines, as they are either read or computed at each detail item. A Blank After field is set to spaces only after the footers for the level break associated with the line on which you specified it are processed. For more information on use of this attribute, refer to the description of the Math Calculation operation in Section 6.6.

As indicated on the screen, touching FN 16 displays a list of defined field names and touching FN 25 deletes the current field.

6.5.4 Special Field Considerations

You can use individual fields from data files as often as you wish, using the same field name at different line or column positions on the report.

You should take care when specifying fields to print in a footer. Because of the order in which the report is processed, a new record is read into a field and overwrites the previous data before the footers for the old record print. If the field in the footer changes from record to record, the print field in the footer line should be different from the record field; you should copy the contents of the record field to the footer field during operation processing.

6.6 OPERATIONS

When you touch FN 02 from the Report Utility Selection screen (Figure 6-3), the Report Operation Specification screen (Figure 6-6) appears.

```
IDEAS Report Generator - Operation Specification              Release 2.1
                                      Specified operations     page 1 of 1
Report program name is MAILREPT
Desc PRINTOUT OF MAILING LIST

Operation name         (number    )
Perform operations only if level >=

Optional test SYSFLAG# =   , State =
If sysflag test is used, operations
are performed only if the level is as
specified AND the sysflag is in the
specified state ("Y" = ON, "N" = OFF)

Number of edits for this operation

//////////////////////////////////////

Touch SF'0 to Add Operation
      EDIT to Modify operation

Touch FN '17-'20 to display pages 1-4
of the reports operation names
EXECUTE = Accept, '0 = Add operation, EDIT = Modify, '09 = Delete, '31 = Cancel
```

Figure 6-6. Report Operation Specification Screen

To add an operation, touch FN 00 and fill in the following fields:

OPERATION NAME
: Each operation has an eight-character name (OPRTNXXX). You can change this name.

NUMBER

As you define each operation, it receives a sequence number. You can change this number by using FN 10.

LEVEL =

You can specify that an operation be performed only if the current level is greater than that specified in this field.

An operation specified as level R (report) takes place both before and after the entire report is executed. If you wish to perform a report-level operation only at the beginning of the file, you should make it conditional based on SYSFLAG1 being ON (refer to the system flag conventions described in Section 6.3). Conversely, if you wish to perform a report-level operation only at the end of the file, you should make it conditional based on SYSFLAG1 being OFF.

For a complete description of processing order in a report, refer to Section 6.8.

SYSFLAG# =

You can specify a system flag test for each operation, depending on whether the flag of the specified number is set to the specified state.

The screen also displays the number of edits previously defined for this operation. If you define more operations than can fit on this screen, touching FN 17 through FN 20 displays additional operations.

When the fields on the screen are full, the message on Line 24 changes to the following:

EXEC=Accept, EDIT=Modify, '16=Edit process, '10=Change op number, '31=Cancel

To define or modify the edits for an operation, touch FN 16 to evoke the Edit Operation Sequence Selection screen. This screen is identical to the Edit Operation Sequence Selection screen of the Screen utilities (Figure 5-11, explained in Section 5.3).

Report operations differ slightly from screen operations. There is no distinction between pre- and post-entry processing. Operations are performed in sequential order for each specified level before the lines of that level print. Additionally, you should note the following differences between screen and report operations:

FN 00 -- PERFORM PASS/FAIL ACTIONS ONLY USING LAST P/F CONDITION

Report utilities do not perform pass/fail actions A, B, C, D, R, U, V, W, X, and Y. For Operation Q, the branch should specify an operation name. Options E and F are valid, allowing you to disable or enable the CANCEL key for an operator during report execution.

FN 01 -- SET FIELD(S) EQUAL TO FIELD(S) AND/OR CONSTANT(S)

You should not set numeric fields to values using this Edit Specification. Use of this Edit Specification for numeric fields can cause erroneous results to occur if later processing within operations requires this numeric field in math calculations, logic tests, or range tests. To set numeric fields to constants or initial values, you should use the Math Calculation option (Option 4).

FN 02 -- READ A RECORD FROM A DATA FILE

You can store full records of up to 150 bytes within a single field defined on a null report line.

Records are read prior to execution of the first operation. Therefore, you should consider the effect that record reads performed during operations will have on records already located in the report buffer.

As discussed in Section 6.3, associated data files that are read automatically provide certain conventions in the use of system flags. If you wish to maintain these conventions when you read a data file with this operation, you must do so by means of the Operation Specification screen and the subsequent Pass/Fail screen.

As discussed in Chapter 5, you should perform a sequential read on only one file at a time during field processing. This restriction also holds for field processing during report execution, except that you can perform sequential processing on one additional file in addition to File 1 without destroying the processing sequence for File 1. If you wish to read more than one additional file sequentially, you must follow the instructions given in Chapter 5.

FN 03 -- PERFORM LOGICAL TEST(S)

This option performs as in the Screen utilities.

FN 04 -- MATH CALCULATION(S)

This Edit Specification should be the only method you use to manipulate numeric fields during report processing. You cannot use @TSTFLDs as part of the math specification. Instead, you should use temporary buffer fields placed on null lines.

You must specify the accumulation of totals and subtotals as a math operation at one level below that for which the total or subtotal prints. For example, addition at the detail level accumulates a Level 0 subtotal. In the case of a report with only one level break (Level 0 subtotals only), you should accumulate the report level total as an addition in a Level 0 operation. While it is possible to accumulate the final total at the detail level, it takes additional time as, typically, many more detail lines are processed than Level 0 lines.

You must specify each subtotal field at a level break with the Blank After attribute on. Otherwise, the subtotal is never cleared and a total accumulates. The Blank After attribute performs its function only when the line actually prints, so if a line does not print for some reason (for example, because of a flag test) the subtotal is not cleared. In this case, you should zero out the field using a conditional operation.

FN 05 -- RANGE TEST(S)
This option performs as in the Screen utilities.

FN 06 -- TABLE LOOK-UP OR TABLE LOOK-UP AND REPLACE
This option performs as in the Screen utilities for alphanumeric fields. You should not use numeric fields as replace fields if you will use them subsequently to perform math calculations.

FN 07 -- USER-SUPPLIED PROCESS
This option performs as in the Screen utilities.

6.7 LEVEL BREAK SPECIFICATIONS

When you touch FN 04 from the Report Utility Selection screen (Figure 6-7), the Level Break Specification screen (Figure 6-7) appears.

```
IDEAS Report Generator - Level Break Specifications                Release 2.1

Report program name is MAILREPT                     Break    Field  File# File     TOF
Description: PRINTOUT OF MAILING LIST               Level 0
                                                    Level 1  CUST #    1 MAILLIST N
No  File      Associated Data File Description      Level 2
1   MAILLIST  CUSTOMERS AND ADDRESSES               Level 3
2                                                   Level 4
3                                                   Level 5
4                                                   Level 6
5                                                   Level 7
6                                                   Level 8
7                                                   Level 9

1. A normal detail line has no level specified (level = blank.)
2. Headers, footers, etc. may be specified for up to 10 levels (0-9), plus
   the report level ("R") for initial headers and final footers.
3. Only fields from associated data files may be used to specify level breaks.
4. A level break will occur whenever the specified field changes in value.
5. When any level break occurs, all lower levels will also be processed.
4. The "TOF" indicator is used to specify a top-of-form (new page) after all
   footers for the specified level have been printed. (Y = go to new page.)

Attention:     EXECUTE = Accept, '0-'9 = Level break specs, '16 = Field names
```

Figure 6-7. Level Break Specification Screen

As indicated on the screen, touching FN 00 through 09 allows you to define the field for the corresponding level break. You must specify the field name and the file number of the file that contains it. Only fields from associated data files can specify level breaks. After specifying a level break, indicate whether a top-of-form occurs after all footers for the specified level print. Section 6.8 gives more information on page breaks.

An example of multiple level breaks follows:

```
 Department 1                              Header line, level 2

    Clerk 1                                Header line, level 1, 1 blank
        Transaction 1                      before detail line
        Transaction 2
           ...
        Transaction N
    Clerk 1 transaction total              Footer line, level 1

    Clerk 2
        Transaction 1
        ...

(New page)

 Department 2
    Clerk 1
        Transaction 1
        ...
```

In this case, a level break of Level 1 occurs when the value of the field that contains the name of the clerk changes. A level break of Level 2 occurs when the value of the field that contains the name of the department changes. Additionally, a top-of-form occurs after all footers for Level 2 print.

6.8 SUMMARY OF PROCESSING OPERATIONS

Processing operations involve the following steps:

1. The application reads the first record.

2. The application performs all report-level operations. If you wish these operations to take place only at the beginning of the report, make them conditional on SYSFLAG1 being ON.

3. The application performs all page-level operations.

4. The page-level headers print.

5. On the first page only, all report-level headers print.

6. All other header lines starting at 9 and going to 0 print.

7. The application performs all detail-level operations.

8. The detail lines print.

9. The application reads the next set of records and makes a level break check. If no level break occurred, the application returns to Step 7. If a level break did occur, the application proceeds to Step 10.

10. The application performs all level operations up to and including the level causing the break.

11. Footer lines for each level print up to and including the level causing the break.

12. The header lines for each level print from the one causing the break to Level 0.

13. The application returns to step 7.

14. At the end of the file, the application performs all level operations, starting at 0 and going to R. Then, all footer lines starting at zero and going to 9 print, followed by all report footer lines. After this, all page-level footers print. If you want to perform an operation at the end of a report only, you should define the operation as a report-level operation conditional on SYSFLAG1 being off.

Page breaks, described below, cause some interruption in the processing sequence. Page breaks can be caused by either of two conditions. The first condition occurs when a level break causes the printer to issue a top-of-form. You can define this during level break specification, described in Section 6.7. Before a page break occurs, all footers of a level less than or equal to the level causing the page break print. Then headers print, starting at the level equal to the level causing the page break and going down to detail level. Fields of all grouping levels print the first time the line in which they are defined appears on a page.

The second condition that causes a page break is when the specified number of lines per page, defined on the Main Report Specification screen (Figure 6-1), is exceeded. If the specified number of lines per page is zero, no page breaks occur except for those caused by level breaks. If the specified number of lines per page is 66, page breaks occur, but the printer does not issue an automatic top-of-form; in other words, all operations that normally occur at a page break occur, but the printer does not issue a top-of-form before printing headers, assuming that the 66 lines have filled the page. You should not specify footers for a 66-line page, since this throws pagination off. For any other indicated number of lines per page, the following applies:

1. The number of lines per page does not include page footers. If you have a 2-line page footer and want 60 lines per page total, you must specify 58 lines per page.

2. A page break prints any page footers. A top-of-form follows, then the page headers, then fields of all grouping levels. This can occur at any point in the detail, footer, or header processing.

3. Level 0 through 9 headers do not print automatically at the top of each page. Column headings that you need on each page must be page header lines.

6.9 RECORD SELECTION

You can print the records of an IDEAS file according to a variety of parameters, and you have the option of allowing the operator to modify any of these parameters at runtime. If you specify a range of records, using minimum and maximum values, only a portion of the file is processed. You can select records to be printed according to as many as three logical tests performed on the fields. You can also determine the order of record processing according to the logical sequence of as many as five fields on the screen.

Touching FN 00 through 03 from the Report Utility Selection screen (Figure 6-3) defines record selection for reports. Touching FN 00 evokes four screens, one after the other (Figures 6-8 through 6-11). The first screen enables you to specify which fields to use for record selection and to assign descriptions to these fields for operator use. The second screen enables you to specify range parameters for record selection. The third screen defines logical tests for fields, specifying which records should print. The fourth screen delineates sort specificatons.

You should keep in mind two aspects of record selection. First, if you specify record selection involving a numeric field and there is a non numeric value in that numeric field (a situation that could result from a designer-specified copy, or from a work buffer that is not cleared when a screen is loaded), the record is always processed. Second, you should never compare alphanumeric fields with numeric values; such a comparison yields inaccurate results.

A screen-by-screen explanation of the record selection screens follows.

Record Selection: Field Description

When you touch FN 00 from the Report Utility Selection screen (Figure 6-3), the Record Selection Field Description screen (Figure 6-8) appears.

```
IDEAS Report Generator Record Selection & Sort Field Descriptions  Release 2.1

Report program name is MAILREPT
Description PRINTOUT OF MAILING LIST

Enter up to 15 field descriptions which may      File #          Run time description
be used to specify range, selection, & sort      ↓ Field         seen by the operator
at run time. (1 reserved for key range spec)     1 CUST #

File #
↓    File     Data file description
1 MAILLIST    CUSTOMERS AND ADDRESSES
2
3
4
5
6
7

'0 Permit run time key range modification? Y
'1 Permit run time selection modification? Y
'2 Permit run time sort spec modification? Y

EXEC=Accept, EDIT=Modify, '09=Delete spec, '16=Field names, 31=Cancel
```

Figure 6-8. Record Selection Field Description Screen

This screen defines which record selection components the operator can modify at runtime. As the screen indicates, FN 00, 01, and 02 serve this function. Only screens indicated as modifiable appear to the operator.

This screen also specifies which fields you can select for use in record selection and sorting. You can select up to 15 fields. One of the seven data files associated with the report must contain all of these fields, but you need not define the fields specifically as fields in the report itself. These fields are the only ones that you can use when specifying record selection and sorting.

Since it is seldom convenient for the operator to use the eight-character field names, this screen allows you to specify more mnemonic field descriptions of up to 20 characters. This description, rather than the field name, identifies the field during subsequent record selection and sorting specifications. This description is also the description that appears at runtime when the operator specifies record selection and sort. The first N field names, where N is the number of fields constituting the key for the sequence file, are reserved for key fields from File 1. You can actually select only 15 - N fields, since the system automatically selects the first N. You cannot change the key fields, but you can specify descriptions for them.

Record Selection: Range Specification

When you touch FN 01 from the Report Utility Selection screen (Figure 6-3), or when you first accept the Record Selection Field Description screen (Figure 6-8), the Record Selection Range screen (Figure 6-9) appears.

```
 Report Range Specification for PRINTOUT OF MAILING LIST          - MAILREPT

Enter Minimum and Maximum values (RANGE) for each field specified below. If
values are specified, only records within that range will be processed. All
others will be ignored. If no values are specified, ALL records will be
processed. FN'09 will cause the current range specification to be discarded.

Even if a key was specified as descending, the "minimum value", of the WHOLE
key, to enter below should be less than, in the usual sense, the "maximum
value". They may of course be equal, or be left blank to process all records.

Range Specification:
Field Description        Minimum value              Maximum value


FN '00    Allow User Modification of RANGE Default Values at Execution ?   Y
EXEC=Accept, EDIT=Modify, '0=User mod, '16=Desc, '31=Cancel
```

Figure 6-9. Record Selection Range Screen

Regardless of the eventual sort specification, a controlling sequence file for a report always exists. This is the file specified as File 1 on the Main Report Specification screen (Figure 6-1). The records are scanned first in key sequence order for this file. The minimum value should always be less than or equal to the maximum value, whether the key is in ascending or descending sort order.

As indicated on the screen, you can limit up to five key fields of this controlling sequence file to minimum and maximum valuea. Only those records whose key falls within the specified range are processed. All other records in the file are not processed in any way. You can specify only one value by leaving the maximum or minimum value blank, as desired. You can permit the operator to modify the range selection at runtime.

Record Selection: Logical Testing

When you touch FN 02 from the Report Utility Selection screen (Figure 6-3), or when you first accept the Record Selection Range screen (Figure 6-9), the Record Logical Testing screen (Figure 6-10) appears.

```
     Report Selection Logic for PRINTOUT OF MAILING LIST              - MAILREPT

You may specify up to 3 tests with any combination     Fields allowed for tests
of AND & OR.  Only those records which pass the          Identifier
tests will be processed.  All others are ignored.      Y    Description
The first entry must an Identifier other than "V"      A =
referring to a description listed at the right.
"Op" may be:  = , <> , > , >= , < , or <=
The second entry may be an Identifier or "V".
If "V" is used, you may enter a constant value to
be tested, rather than a field.

Record Selection Specification:
  Identifier                         Identifier
Y Description                  Op  Y Description or Value

                 (AND/OR)

                 (AND/OR)
                                                       V = (Any constant value)
FN'0    Is LOGIC User Modifiable at Execution ?   Y

EXEC=Accept, EDIT=Modify, '0=User mod, '9=Delete spec, '16=Range, '31=Cancel
```

Figure 6-10. Record Selection Logical Testing Screen

As this screen indicates, you can specify up to three logical tests for any of the fields described on the Record Selection Field Description screen (Figure 6-8). The system processes only those records whose fields pass the specified tests. The complete test can be any combination (AND/AND, AND/OR, OR/OR, OR/AND) of up to three individual tests.

For each test, specify the following items:

1. Which field to test, from the list of up to 15 descriptions. You indicate the field by keying in a one-letter identifier (listed on the screen in a table with the 20-character field description). The test specification then displays the description.

2. The logical operator (=,<> , > =, < , < =, >)

3. Which value to test against. This can be either a field from the list of descriptions, which you indicate by its identifying letter, as in the first field specified, or a constant value, which you indicate by keying in the letter V and the value to use, which can be up to 20 characters. If you test the field against a constant value, the specified constant has the same attributes as the field from the associated data file.

You can permit the operator to modify the logical record selection specification at runtime.

Record Selection: Sort Specifications

When you touch FN 03 from the Report Utility Selection screen (Figure 6-3), or when you first accept the Record Selection Logical Testing screen (Figure 6-9), the Record Selection Sort Specification screen (Figure 6-11) appears.

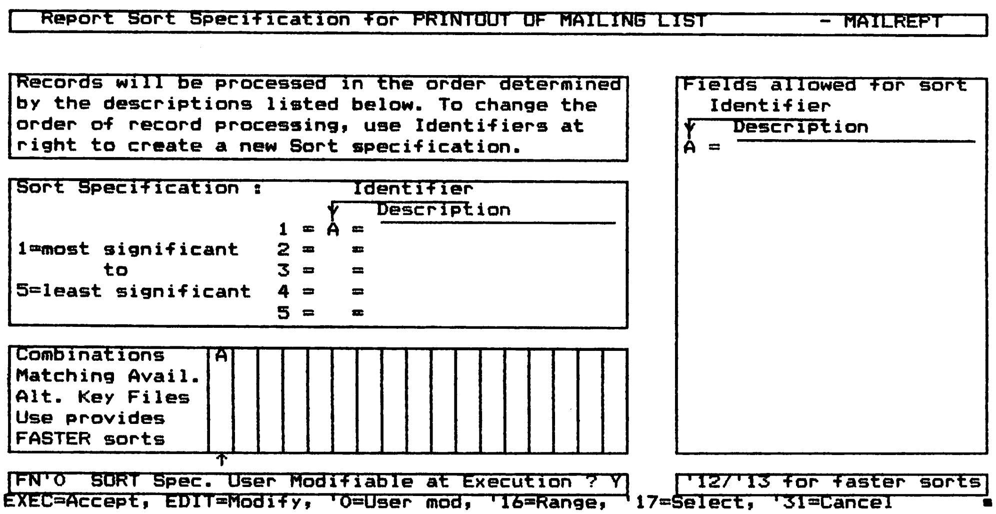


Figure 6-11. Record Selection Sort Specification Screen

This screen allows either you to specify that the records of the report prints according to a logical sequence other than that of the key fields of the sequence file, File 1 (as specified on the Report Generator Main Specification screen, Figure 6-1). Touch EDIT to specify fields as sort criteria, in their order of importance. You then select any combination of the fields described in the Field Description screen (Figure 6-8), which appear along with their identifiers on the left side of the screen.

The sort specification consists of up to 5 fields from the list of 15 descriptions. The default sort specification is the key construction of the sequence file.

Although you can use any previously described fields to determine the sort order on a report, processing time decreases if the fields are alternate key fields. Because of this, during development, fields that are also key fields appear in the box in the lower right corner of the screen. By touching FN 11 and 12, you can move an indicator arrow to the left and right and the appropriate identifier combinations default. The field combination currently indicated automatically appears as the current sort specification.

You can permit the operator to modify the sort specifications at runtime. This feature can cause difficulty if you use it in conjunction with level breaks. For example, assume that an inventory report is printing parts and vendors. You can design this report to process the file in vendor number/part number order, with a user-modifiable sort specification so that the operator can also print the report in vendor name/part number order. A level break on vendor number would show subtotals by vendor. If, however, the operator sorts the report by part number, the level breaks still occur every time the vendor changes and the resulting report can be inappropriate. Because of this difficulty, you should use operator-modifiable sort primarily for lists; developer-designed reports that use extensive level breaking should not have a modifiable sort.

Using the sort function requires a sort work file. The first time the sort module is run, the operator must specify the size of the sort file that the system will create. If the sort module is run again with a different sort order specified, the system might require a larger sort file. When this occurs, a message asks the operator to respecify the size of the sort module. Section 6.11 describes the sequence of screens that appears to the operator.

6.10 CREATING A REPORT PROGRAM

After you define all components of the report, return to the Report Utility Function Selection screen (Figure 6-3) and accept the screen. IDEAS creates the control file for the report program and a message appears indicating that you can generate the report program itself or cancel to the Report Utility menu. When you first create a report, you must define the report program. If you are revising the report, however, you must regenerate the program only when you change the Edit Specifications or add associated data files.

6.11 REPORT EXECUTION: RUNTIME ORDER AND APPEARANCE

When an operator runs an IDEAS report, the following screens appear:

1. If record selection range is modifiable at runtime, the Record Selection Range screen (Figure 6-9) appears.

2. If record selection logical testing is modifiable at runtime, the Record Selection Logical Testing screen (Figure 6-10) appears.

3. If sort specifications are modifiable at runtime, the Record Selection Sort Specification screen (Figure 6-11) appears. The listing of faster sort combinations in the bottom right corner of Figure 6-11 does not appear to the operator.

   After the Sort Specification Screen appears, the next screen that appears depends on whether this is the first time a sort specification is being defined.

   a. If this is the first time any sort specification is being defined, the operator must create a sort work file. The screen displays a default address for this work file that is the address of the disk specified in the START module with the most free space. The operator can change this address. The screen displays how many sectors the sort file requires; the operator can accept this size or indicate a larger size (anticipating future use of the sort file for larger files). The operator then accepts the screen.

   b. If a sort file that is large enough to handle the sort already exists, no operator input is required and control goes to the next screen.

c. If a sort file already exists that is not large enough to handle the sort, a message indicating this appears to the operator, along with the address of the disk containing the sort file and how many sectors the new file requires. The operator can do one of three things:

   1) Cancel and return to the main menu.

   2) Create a new sort file (described in Step a above).

   3) Accept the file, even though it may not be large enough to accommodate all of the records. This might have no effect on the report if logical selection will eliminate many of the records from the report.

4. If the printer device number is modifiable, a screen appears indicating the name and description of the report to be printed, followed by a field in which to modify the device address for the desired printer. The default device addresses, printers, and fonts established with the Manage System Peripherals utility appear, as described in Subsection 2.3.3.

5. A screen appears indicating that the sort and selection process is being performed.

6. After the operator touches EXECUTE, any defined test lines are executed, followed by any indicated banner lines.

7. The report then begins, with page level header lines. Refer to Section 6.5.

If neither record selection nor device selection is user-modifiable, if a sort file already exists, and if the device address is selected to 005 (the CRT), you can load the report program as part of screen processing using Pass/Fail Option S. The report then prints on the operator's screen, one screenful at a time. This in effect makes the report generator an extension of the screen processor. If you do this, and the report processing then loads a screen module, you must specify a COMCLRPT (clear report variables) on the load, as explained in Section 5.5 under Pass/Fail Option S.

CHAPTER 7
MENUS

7.1 OVERVIEW

In a typical IDEAS application, the START program loads a main menu which evokes screen programs, report programs, other START programs, and submenus. The Menu Program utilities create, revise, and document the menus of an IDEAS-generated program. Function keys or SPACE/BACKSPACE select activities from menus created through the Menu Program utilities. A box to the left of the highlighted entry name indicates the current system default, loaded by pressing EXECUTE or RETURN. In short, a menu created using the IDEAS Menu utility operates in the same way as the menus of the IDEAS utilities themselves.

You can call up to 17 submenus or programs for each menu. Each menu entry has an associated user class that specifies the minimum user class for the program. If a user running an IDEAS-generated menu is of a lower user class than a program requires, that program does not appear on the menu to that user at runtime. Each menu entry can also have an optional user ID code associated with it that restricts its use to the specified user. Any programs with a user ID different from that of the current user do not appear on the menu at runtime. In addition, the developer can specify a different password for each program on a menu. An example of a complete menu compared to the same menu as it appears to a user of limited user class appears in Section 7.3.

When defining a menu, the developer can specify the program type of the programs the menu lists. This allows the developer to indicate whether the program is an IDEAS program, a program from another Wang Utilities system, or any other 2200 program.

The developer can include the IDEAS Release 2 Development Utilities menu as an entry on an application menu screen, allowing direct movement from an application menu to the IDEAS Release 2 Development Utilities menu. This facility allows an operator to modify applications during runtime and greatly facilitates testing of applications during the development cycle.

7.2 MENU CREATION

To evoke the Menu Utility screen, Figure 7-1, touch FN 12 at the Primary Program Selection screen. The four main Menu Utility options, Create, Revise, Document, and Document (Batch), operate as they do in the other utilities.

```
IDEAS Application Menu Editor - Utility Selection Menu Module        Release 2.1

                         FN Program Module
                       ■ 00 Create
                       • 01 Revise
                       • 02 Document
                       • 03 Document (batch)
                            -- Other Functions --
                       • 06 Data files
                       • 07 Screens
                       • 08 Programs (interactive)
                       • 09 Programs (batch)
                       • 11 Reports
                       • 13 START programs
                       • 14 Other utilities
                       • 15 Applications
                       • 31 Main IDEAS menu

SEP 27 83                                                         Term #  2
Mem 36.00                   Global Selected                       Sta. # 20
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 7-1. Menu Utility Screen

To create a menu, touch FN 00, and enter the name of the menu you want to create. The Menu Specification screen (Figure 7-2) appears.

```
IDEAS Application Menu Editor - Menu Specification Module          Release 2.1
Menu name MAILMENU  Rev no     Revised 09/27/83 by SJL  Enter menu title below
■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
Version   Application     Function      User class   UID      Password
FN Program  Description (for display)        Program type (I,W,X) UC UID Password
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15                                                              (CANCEL entry)
16
```

Figure 7-2. Menu Specification Screen

The Menu Specification Screen appears with the menu name, revision number, currently entered date, and current user ID in fields along the top of the screen. The cursor at the description field. Enter a description of the menu. This description appears at the top of the menu. The next line of fields includes:

VERSION, APPLICATION, FUNCTION

Use these three fields for organization of the menus, for batch documentation, and for application installation. Refer to Section 2.2.

USER CLASS

This field defaults to the currently entered user class. The user class determines the minimum user class required for editing and documentation privileges for the menu.

USER ID (UID), PASSWORD

These fields restrict editing and documentation to a particular User ID, or require a password.

After you enter or bypass the above fields, define the menu itself. At each function key you want to use in the menu, enter the following information:

PROGRAM

The program file name of the program the associated function key calls. The program file name can be the name of any IDEAS program, including report programs, other menu programs, and START programs. You may want to separate components of an application into several START programs, so that only selected data files are open for any particular section.

DESCRIPTION

The 51-character description that appears as the load message on the menu for the associated program. Only the load message, and not the actual program name, appears on the menu alongside a function key.

PROGRAM TYPE (I, W, X)

I -- A Type I program is an IDEAS program. This type of program consists of one of two categories:

1. IDEAS-generated: any program, menu, START module, batch program, or report created and/or revised using the IDEAS Release 2 System utilities.

2. IDEAS-compatible: any designer-coded program starting at line 1000; it can use IDEAS Release 2 variables and defined functions.

W -- A program from another Wang Utilities system, such as Datamerge or 2200WP. When you load a Type W program, system variables such as System Date and User ID remain unchanged.

X -- Any other 2200 program. Loading a Type X program clears all variables, then loads the specified program.

If you leave this field blank, IDEAS assumes a Type I program.

UC (USER CLASS)

This field specifies the minimum user class (0 through 9, A through F) to which the specified program is available. If a user running an IDEAS-generated menu has a user class lower than the one indicated here for a program, that program does not appear to that user at runtime.

Class 0 appears as the default value for this field, regardless of the class specified for execution privilege when you defined the program. You can change this to any class. The class you indicate here only determines whether the entry appears on the menu at runtime; whether the operator can execute the program depends on the class indicated for execution privilege. For example, for a program with an execution privilege of C that you assign a user class of 0 on the menu, an operator of classes 0 through B sees the entry on the menu, but receives an error message if he or she attempts to select that entry.

UID (USER ID)
This field specifies the optional user ID code that provides exclusive use of the program to the specified user. Any programs with a user ID different from that of the current user do not appear on the menu at runtime.

PASSWORD
You can associate a password with each program on the menu except CANCEL. If you do not specify a password for a particular program, that program does not require password entry. Each program can have a different password.

CANCEL
An IDEAS-generated menu includes the option of touching FN 31 to evoke any the module you specify.

7.3 EXAMPLES OF RESTRICTED USER ACCESS

The following screens demonstrate restricted user access. Figure 7-3 shows a sample screen during development.

```
IDEAS Application Menu Editor - Menu Specification Module              Release 2.1
Menu name DEMOMENU  Rev no   0 Revised 09/27/83 by SJL  Enter menu title below
EXAMPLE OF RESTRICTED USER ACCESS
Version 1 Application AAA Function DEMO User class 0 UID       Password
FN Program   Description (for display)          Program type (I,W,X) UC UID Password
00 SECUPROG SECURITY ADMINISTRATION                                 I F
01 PAYROLL  PAYROLL ADMINISTRATION                                  I E
02
03 UPKEEP   ADD NEW EMPLOYEES                                       I 9
04 PRNTMENU LIST OF AVAILABLE REPORTS                               I 5
05 PRINTEMP PRINT A LIST OF EMPLOYEES                               I 0
06
07 GAMES    MENU OF AVAILABLE GAMES                                 X 0 SJL
08
09
10
11
12
13
14
15
16                                                            (CANCEL entry)
Attention:      EXEC=Accept, EDIT=Modify, FN'0-FN'16=Edit a line, FN'31=Cancel ■
```

Figure 7-3. Screen Appearance During Development

When operator SJL, who is of user class F, runs the START program that calls this menu, the menu shown in Figure 7-4 appears at runtime.

```
                    EXAMPLE OF RESTRICTED USER ACCESS



                    FN Program Module
                  ■ 00 SECURITY ADMINISTRATION
                  • 01 PAYROLL ADMINISTRATION
                  • 03 ADD NEW EMPLOYEES
                  • 04 LIST OF AVAILABLE REPORTS
                  • 05 PRINT A LIST OF EMPLOYEES
                  • 07 MENU OF AVAILABLE GAMES


SEP 27 83                                                       Term #  2
Mem 36.00               Global Selected                         Sta. # 20
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 7-4. Runtime Menu for Designated User of Class F

To a user of class F who is not SJL, the menu shown in Figure 7-5 appears in the center of the menu screen.

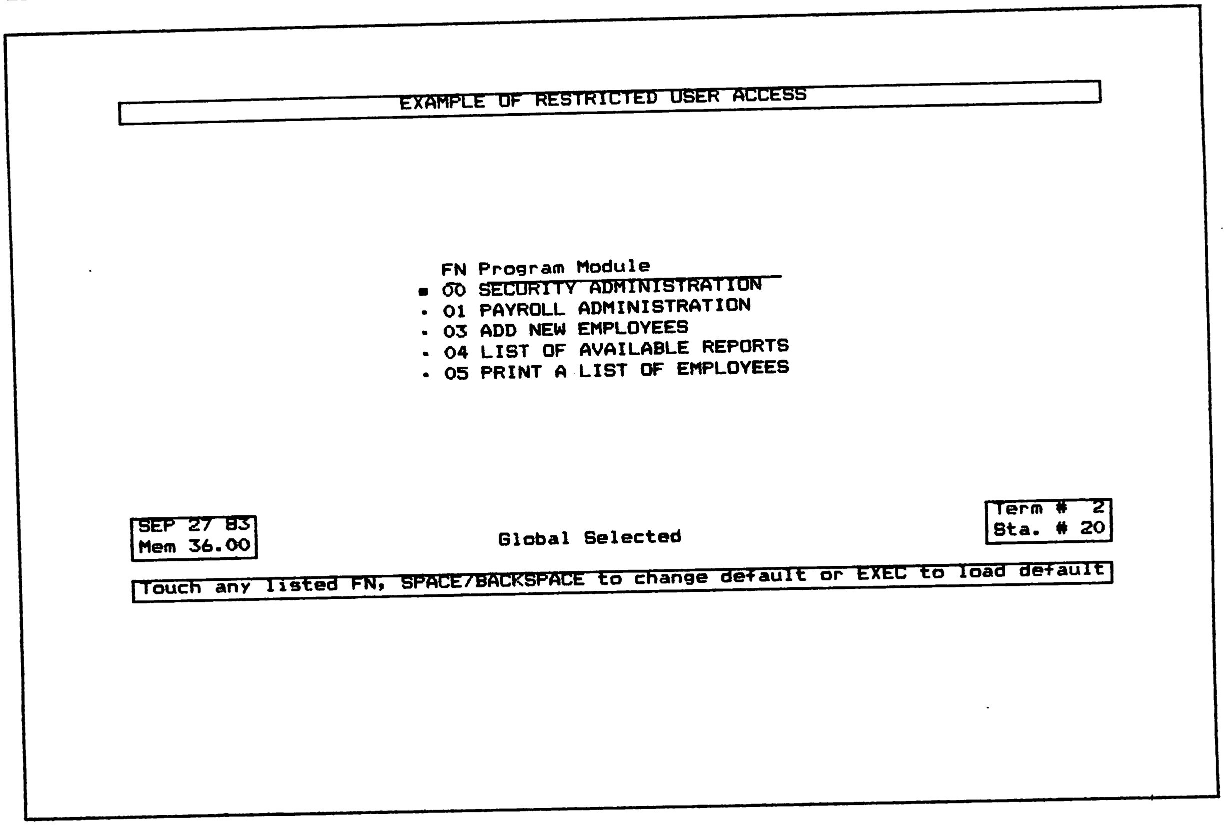


Figure 7-5. Runtime Menu for Other Class F Users

To a user of class 5, 6, 7, or 8, the menu shown in Figure 7-6 appears in the center of the menu screen.

```
                    EXAMPLE OF RESTRICTED USER ACCESS

                         FN Program Module
                       ■ 04 LIST OF AVAILABLE REPORTS
                       · 05 PRINT A LIST OF EMPLOYEES

SEP 27 83                                                    Term #  2
Mem 36.00                 Global Selected                    Sta. # 20
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 7-6. Runtime Menu for User Classes 5 through 8

7.4 REVISION, DOCUMENTATION, DOCUMENTATION (BATCH)

To revise a previously created menu, touch FN 01 at the Menu Utility screen (Figure 7-1), enter the name of the menu to revise, and proceed as if creating a new menu.

To document a menu, touch FN 02 at the Menu Utility screen and enter the name of the menu to document. The Menu Specification screen (Figure 7-2) appears with the following message on Line 24:

EXEC=Full Documentation, FN 0-FN 15=document for user class 0-F, FN 31=CANCEL

As the message indicates, FN 00 through 15 correspond to user classes 0 though F. Touching one of these function keys documents menu entries of that user class or lower. For example, touching FN 04 documents only menu entries of user classes 0, 1, 2, 3, or 4.

To document a series of menus at once, touch FN 03 at the Menu Utility screen. The Menu Batch Documentation screen (Figure 7-7) appears.

```
IDEAS Application Menu Editor - Menu Batch Documentation Module     Release 2.1

Note:   This module will document all menu program
        files which meet the following criteria:

     1. A password is not required for documentation
     2. The edit/document user class is <= yours
     3. The user ID, if any, is yours
     4. The version, application code, and function code
        are as specified below.  A blank entry in any of
        these 3 will cause that parameter to be ignored.

          Version              =        .

          Application code     =

          Function code        =
```

Figure 7-7. Menu Batch Documentation Screen

The screen lists the criteria the module uses in documenting the menus.

CHAPTER 8
START PROGRAMS

8.1 OVERVIEW

The START Program utility creates and revises a module that initializes an application. This module dimensions all of the necessary common variables to the appropriate size for the specified files, sets certain flags, and opens the specified data files. In addition, it loads the system subroutine module, the System Date module, and the specified application module (normally, but not necessarily, a menu module).

There are two major sections of this utility. The first section delineates the general specifications for the START program you want to generate, including security considerations, peripheral device assignment, and the program module that the START program loads. The second section of the utility indicates which data files to open.

8.2 START PROGRAM CREATION

To evoke the START Program utility, touch FN 13 at the Primary Program Selection screen. The START Utilities menu appears, indicating that you can create, revise, or document a START program. When you choose the creation option, a screen appears and requests a name for the module you are creating. Once you enter a name, the START Module General Specification screen (Figure 8-1) appears.

```
IDEAS "START" Module Editor - General Specifications                Release 2.1
Module name   MAILSTRT                              Revision number          By SJL
Description   ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■      Last revision date    09/27/83

Version                  Edit & document privilege   Application access privilege
Application              User class    User ID       User class    User ID code
Function                 Password                    Password

Peripheral device address assignments  #08/ D21 Application data files or other
#01/ 216 Printer                       #09/ D21 Application data files or other
#02/ D23 IDEAS system utilities        #10/ D21 Application data files or other
#03/ D21 Screen & Report masks         #11/ D21 Application data files or other
#04/ D21 Application programs          #12/ D21 Application data files or other
#05/ D21 Data file control files       #13/ D21 Application data files or other
#06/     Telecommunications board      #14/ D21 Application data files or other
#07/ D21 Application data files        #15/ D21 Application data files or other

Program module to load in case of access privilege or other failure =
Program module to load under proper, normal execution circumstances =
Set @SYSBUFO=
FN'16 Work buffer size    ( 1750)  1750   FN'18 Sort buffer size  ( 256)   256
FN'17 Record buffer size  (  256)   256   Number of primary files opened =   0
```

Figure 8-1. START Module Specification Screen

Complete the following fields:

DESCRIPTION

Enter a description of the program in the Description field. This description appears at the top of the screen when the application is being loaded.

VERSION, APPLICATION, FUNCTION

Refer to Section 2.2.

EDIT & DOCUMENT PRIVILEGE

Enter the minimum user class, user ID, or password required to edit or document the START module.

APPLICATION ACCESS PRIVILEGE

Enter the minimum user class, user ID, or password required to run the START module.

PERIPHERAL DEVICE ASSIGNMENTS

Enter the device addresses for the developed application. These addresses default to the addresses set at the Peripheral Device Selection screen (Figure 2-6).

PROGRAM MODULE TO LOAD IN CASE OF ACCESS PRIVILEGE OR OTHER FAILURE

If the program that the START module loads after initializing the application is not available to the current user, the program module specified here is loaded. Some possible reasons for access privilege include an invalid user ID for program execution, an invalid user class for program execution, or a system disk access error. If you leave this field blank, failure to load the program results in an error message, and execution stops.

PROGRAM MODULE TO LOAD UNDER PROPER, NORMAL EXECUTION CIRCUMSTANCES

Enter the program that the START module loads after initializing the application.

SET @SYSBUF0=

The system buffer, @SYSBUF0, is described in Subsection 4.2.1. @SYSBUF0 allows you to establish a 64-character string that is available without requiring a disk read and that remains constant from menu to menu. You can change the value of the system buffer within the application.

FN 16 -- WORK BUFFER SIZE

This field originally defaults to its minimum value of 1750 bytes. After entering the data files, the work buffer defaults to the sum of the seven largest data files. You can decrease this number, but it cannot go below 1750. If any screen or report requires a larger work buffer, you must set it here in the START module. You must note the required size of the work buffer indicated on the Screen Mask Specification screen (Figure 4-3) and the Main Report Specification screen (Figure 6-1).

FN 17 -- RECORD BUFFER SIZE

The record buffer size defaults to the largest record size of all files opened by the START module. You can change this figure, but the record buffer size cannot go below its minimum of 256 bytes.

FN 18 -- SORT BUFFER SIZE

The size of the sort array of the largest file the START module opens that is used for sequential processing determines the sort buffer default size. You can change this figure if, for example, no sequential processing will be performed on the largest file. The sort buffer size cannot go below its minimum of 256 bytes.

NUMBER OF PRIMARY FILES OPENED

A START module can open up to 60 primary data files, and each primary data file can have up to 16 associated alternate key files. This field indicates how many files are already open.

After you enter the fields that have no default values, a prompt appears at the bottom of the screen instructing you to accept the screen as is or to modify any of the fields on the screen. After you accept the screen, the START Module Data File Specification screen (Figure 8-2) appears. This is the screen on which you indicate which data files the START module opens.

```
IDEAS "START" Module Editor - Data File Specifications
Total number primary and alternate files open =                         Release 2.1
FN   File    Description                          Use Sort  Primary files selected
--   ------  (Primary file)                       --- ----
00

             (Alternates key files if any)
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
Use codes S=Sequential/random R=Random only N=Not used   Sort=Sort buffer size
EXEC=Accept   'O =Add a file  FN'15=Help FN'31=Cancel
```

Figure 8-2. Start Module Data File Specification Screen

As the prompt at the bottom of the screen indicates, you can accept the screen by touching EXECUTE, add or edit a file by touching FN 00, delete a file by touching FN 09, evoke a HELP screen that explains the use codes (see below) by touching FN 15, or cancel out of the screen by touching FN 31.

When you touch the EDIT key to direct the system to open a file in the START module, the name and description of the primary file appear on the screen; the names and descriptions of the associated alternate files also appear on the screen. After you edit the primary and associate files, one of the screen columns on the right half of the screen contains the name of the primary file only.

When you edit a file in the START module, the system displays the following information for the primary file and all of the alternate files:

File name
Description (32 bytes, assigned by the user at file creation)
Use code (N, R, or S, as described below)
Sort buffer size required if this file is to be processed sequentially.

The use codes are:

S

You can use the file for both random and logically sequential operations. This is the most common use code.

R

You should use an R file for random access only. You can use an R file for add, change, and delete operations. It is possible to use an R file for sequential processing if there is also an S file open for this application that has a sort buffer requirement large enough to accommodate sequential processing of the R file. However, if no specified S file open for this application has a large enough sort buffer, and the sort buffer requirement for the R file is greater than 256 bytes, a fatal error occurs.

N

The file cannot be opened. This has slightly different meaning for primary and alternate files.

1. For alternate files, the START module does not open the particular alternate key file; it opens the primary file and any other alternates not specified as N. You can perform no maintenance on the file's key fields, even though the primary file or another alternate file can change these fields.

   <u>Use this attribute with extreme caution!</u> If, for example, the primary key or another alternate key that is open retrieves a record for update, and the key for this particular file is changed in the record, the alternate key file is not maintained for any alternate file that is not open. While this seems to present potentially major problems at first, the ability to not open some alternate key files can be of great benefit in a fixed-partition environment. For example, if an application has a large customer or employee master file and an occasional report, keyed by name, is needed, an IDEAS program can open just the primary key and the "name" alternate key, even though there may be other alternate keys.

2. For primary files, a use code of N causes the open process to ignore primary and all alternates even though some of the alternates may have S or R specifications, because you cannot open an alternate without its corresponding primary file. This allows you to ignore certain files temporarily (usually for testing purposes) while still associating them with the START module. You can respecify them later without entering the files again.

Accepting the Data File Specification screen returns the START Module Specification screen (Figure 8-1). You can change buffer sizes at this point.

8.3 START PROGRAM REVISION AND DOCUMENTATION

To revise a previously created Start Program, touch FN 1 from the START Utilities menu, enter the name of the menu you want to revise, and proceed as if creating a new START Program.

To document a START Program, touch FN 02 at the Menu Utility screen and enter the name of the menu you want to document. The START Module Specification screen (Figure 8-1) appears with a message on Line 24 indicating that you can touch EXECUTE to print START file documentation.

CHAPTER 9
INTERACTIVE PROGRAM GENERATION

9.1 OVERVIEW

The Interactive Program Generator creates a program according to one of two procedures. In the first procedure, the Program Generator bases its program solely on the Edit Specifications and pass/fail options defined in the Screen Mask Editor. Section 9.2 describes this procedure. In the second procedure, the Program Generator automatically creates a simple program according to certain standard prototypes. Section 9.3 describes the requirements for this procedure and presents descriptions of the automatically-generated programs. Section 9.4 describes the Protect option of the Program Generation menu.

9.2 PROGRAM GENERATION USING EDIT SPECIFICATIONS

To evoke the Interactive Program Generation menu, touch FN 08 at the Main Menu screen. The two Interactive Program Generation Utility options, CREATE and REVISE, operate as they do in the other utilities.

To create a program, touch FN 00 and enter the name of the program you want to create after the next screen appears. The Program Specification screen (Figure 9-1) appears.

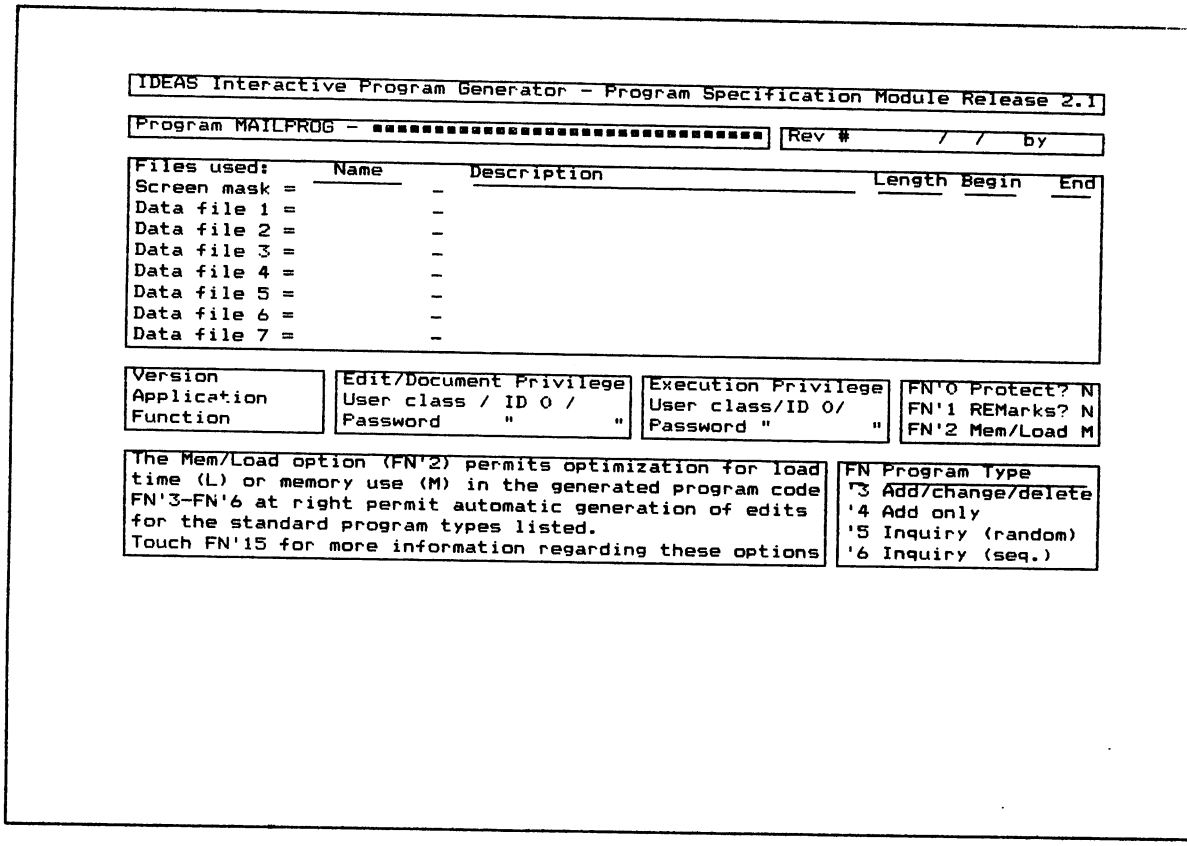


Figure 9-1. Program Specification Screen

The Program Specification screen appears with the cursor under the description field. Enter any convenient program description.

The cursor moves to the next field (the Screen Mask field). Enter the name of the screen mask that the program is to call. The screen mask description and the names and descriptions of the associated data files appear.

The next fields (the Version, Application, and Function fields; the edit/document fields; and the execution privilege fields) serve the same function as they do in all other main specification screens.

The next field, Protect, allows you to specify whether the generated program is to be protected against modification. If you enter Y, you can modify the program only by modifying the Edit Specifications through the screen utilities and then regenerating the program. You should not modify an IDEAS program manually; doing so may cause IDEAS to fail to recognize the program as an IDEAS program. Even after you accept this field and the "Touch EXECUTE..." message appears at the bottom of the screen, touching FN 00 changes the indicated attribute. It is possible to protect a program after you have generated it; refer to Section 9.4 for instructions.

The next field, REMs, determines whether the generated program includes internal documentation. Even after you accept this field and the "Touch EXECUTE..." message appears at the bottom of the screen, touching FN 01 changes the indicated attribute.

The next field, Mem/Load, allows you to choose whether you wish the generated program to optimize for memory or for speed. If you wish to save memory, indicate M; this causes the generated program to overlay its own code in sequence, a procedure that saves space in memory but requires some time. If you wish to save time, indicate L; this causes every section of the program to be loaded into memory at once, a procedure that saves execution time but requires more space in memory.

At this point, you can accept the screen to generate a program according to the Edit Specifications you defined for the associated screen, or you can touch FN 03, 04, 05, or 06 to generate the edits for a standard program automatically. For instructions on implementing this feature, refer to Section 9.3.

If you wish to generate a program based on your own Edit Specifications, accept the screen and IDEAS generated the program. Messages appear on Line 24 that indicate which sector of the program is being processed. When processing is complete, the Program Generation menu appears. You can create another program or proceed to another IDEAS module.

Once you generate a program for a screen, you can archive the Edit Specifications for that screen from the Edit Specification file (using the Extra Screen utilities, described in Section 4.2.3), because they are incorporated into the program itself. The Edit Specification file is not used when an application is run. If you wish to modify the Edit Specifications of a screen after you have generated a program and you have archived the Edit Specifications, you must first restore the Edit Specifications from archive and, after you modify them, regenerate the associated program.

9.3 AUTOMATIC PROGRAM GENERATION

As the Program Specification screen indicates, there are four types of standard programs for which you can automatically generate field edits. The automatic edits only apply to the data file specified as File 1 on the Screen Mask Specification screen (Figure 4-3).

Automatic program generation requires that the screen used for the program fulfill the following conditions:

1. The screen must contain all of the key fields that constitute the data file specified as File 1 on the Screen Mask Specification screen.

2. The fields on the screen must have the same names as in the data file.

3. You have defined no Edit Specifications for the key fields of the associated file.

4. If there is a field at Row 24, Column 80, it must have no associated Edit Specifications. If no field exists at this position, the Program Generator creates one.

To generate one of the standard programs, touch the appropriate special function key at the Program Specification screen (Figure 9-1). All of the programs establish the appropriate Edit Specifications for the key field(s) of the data file specified as File 1 on the Screen Mask Specification screen and for the acceptance field at Row 24, Column 80. You can select one of the following programs:

FN 03 -- ADD/CHANGE/DELETE
This program permits the operator to add, modify, and delete records to the data file.

FN 04 -- ADD ONLY
This program permits the operator to add new records to the data file, but does not allow the operator to delete or modify records.

FN 05 -- INQUIRY (RANDOM)
This program permits random retrieval and display of the records in the data file, but does not permit the operator to add to, modify, or delete records.

FN 06 -- INQUIRY (SEQ.)
This program permits sequential retrieval and display of the records in the data file, but does not permit the operator to add to, modify, or delete records.

After you select the appropriate function key, the Automated Edit Generation screen (Figure 9-2) appears.

```
IDEAS Interactive Program Generator: Automated Edit Generation      Release 2.1

              Program   AUTOPROG  AUTOMATIC PROGRAM GENERATION
                        Program type selected = Add/change/delete

              File      MAILLIST  CUSTOMERS AND ADDRESSES
                        Data file's first key component     CUST #
                             key component 2 if required
                             key component 3 if required
                             key component 4 if required
                             key component 5 if required

              Screen    AUTOSCRN  MAILING LIST OF CUSTOMERS
                        First field on the screen           CUST #
                        Acceptance field @ row 24, col 80   ENDFIELD
                        Data file's first key component     CUST #
                             key component 2 if required
                             key component 3 if required
                             key component 4 if required
                             key component 5 is required

              Edit file is  77% full.          498 records available

Attention:        Touch EXEC to accept,  EDIT for screen, FN'31 to cancel
```

Figure 9-2. Automated Edit Generation Screen

IDEAS checks to see that the requirements for automatic generation are met. If they are not met, you can touch FN 31 to cancel out of the module, or the EDIT key to return to the Screen Mask Editor and make the appropriate changes to the screen mask. If the conditions are met, touching EXECUTE creates the Edit Specifications appropriate to the selected program type in the Edit Specification file.

After you create the Edit Specifications, you can touch EXECUTE again to generate the program now, or EDIT to return to the Screen Mask Editor. If you return to screen definition, you must return to this module to generate the program in the manner described in Section 9.2. The Edit Specifications for the screen are now in the Edit Specification file, just as if you had defined them manually through the Screen utilities.

If you intend to define additional Edit Specifications for the automatically-generated programs, keep in mind the following conventions in the use of system flags. The programs automatically generated through this feature use System Flags 0 and 1; do not use them in your additional Edit Specifications. In addition, the programs set System Flags 2 through 7 to OFF in the generated pre-entry for the first field if the corresponding associated data files exist in the screen applications.

9.4 PROTECT

If you wish to protect the program from modification after you generate it, touch FN 02 from the Interactive Program Generation menu. A screen appears that requests the name of the program file to protect. Enter a valid IDEAS program file, and the system protects the file and returns you to the Program Generation menu.

CHAPTER 10
BATCH PROGRAM GENERATION

10.1 OVERVIEW

The batch program generation facility of the IDEAS utilities defines a series of operations on a data file that do not require operator intervention. Using IDEAS to define a batch processing program is almost identical to using IDEAS to define a report, which is, in a sense, a batch processing program that allows printing of data. Thus, Chapter 6 provides most of the information you need to use the Batch Program Generation module. This chapter presents the batch processing screens and explains the sections of the utility that differ from the Report Creation utility.

10.2 BATCH PROGRAM MAIN SPECIFICATION SCREEN

To create a batch program, choose Option 9 from the Primary Program Selection menu (Figure 2-4). The Batch Program Generator main menu appears; this menu allows you to create, revise, or document a batch program. Touch FN 00 and enter the name of the program you want to create. The Batch Program Main Specification screen, Figure 10-1, appears. Except that it contains no provision for specifying number of lines per page and number of line formats, this screen is identical to Figure 6-1, the Main Report Specification screen. Refer to Section 6.1 for information on this screen.

```
IDEAS Batch Program Generator - Main Specification Screen          Release 2.1

Module name     BATCHPRO                          Revision number
Description     ********************************  Last revised 09/27/83 SJL

Version           Edit & Document Privilege       Execution Privilege
Application       User class     User ID          User class     User ID
Function          Password                        Password

       Data files to be used in the module        Current record selection:
FN      File         Description                  Range    User modifiable
01                                                Logic    User modifiable
02                                                Sort     User modifiable
03                                                 by:
04
05
06
07

08  Save program REMs    N   Number of files    0   Number of fields        0
                             Work buffer     1750
10  Program protection   N   Record buffer    256   Number of operations    0
```

Figure 10-1. Batch Program Main Specification Screen

The first time you accept the Batch Program Main Specification screen, the File Reading Key Specification screen, Figure 10-2, appears. When you are revising a batch program, the Batch Generator Function Selection screen, Figure 10-3, appears directly.

```
IDEAS Batch  Generator - Additional Data File Key Specifications    Release 2.1

Batch  program name is BATCHPRO  Description: BATCH PROCESSING

#   File   Associated data file description Comment
1 MAILLIST CUSTOMERS AND ADDRESSES          File is read automatically
2                                           Associated data file 2 is not used
3                                           Associated data file 3 is not used
4                                           Associated data file 4 is not used
5                                           Associated data file 5 is not used
6                                           Associated data file 6 is not used
7                                           Associated data file 7 is not used
For each file used (2-7) specify the construction of the  random access key by
using a field name and file no. to identify a field in a lower-numbered file.
Specify under "I/U" if the file is to read in inquiry (I) or update (U) mode.
FN   File    I/U #fields Field/File Field/File Field/File Field/File Field/File
'1 MAILLIST  I       File is read automatically
'2                            /          /          /          /          /
'3                            /          /          /          /          /
'4                            /          /          /          /          /
'5                            /          /          /          /          /
'6                            /          /          /          /          /
'7                            /          /          /          /          /
EXECUTE = accept,   FN'0 = field names,   FN'1-'7= file specs,   FN'31 = cancel
```

Figure 10-2. File Reading Key Specification Screen

The File Reading Key Specification screen (Figure 10-2) is identical to Figure 6-3. Refer to Section 6.3 for explanation of the screen.

After you accept the File Reading Key Specification screen, the Batch Generator Function Selection screen, Figure 10-3, appears.

10.3 BATCH GENERATOR FUNCTION SELECTION

```
IDEAS Batch Generator - Summary & Function Selection Module          Release 2.1

Batch processing program name is:   BATCHPRO
Description BATCH PROCESSING

      File    Description
1  MAILLIST  CUSTOMERS AND ADDRESSES              ---- Summary ----
2                                                Number of fields            0
3                                                       alpha                0
4                                                       numeric              0
5                                                Number of operations        0
6                                                Number of level breaks      0
7

FN Specification Options      Used  Modifiable   FN Specification Module   Used
'0 Field descriptions:        No    @ run time   '5 Fields                  No
   Record selection (Range)   No       No        '6 Operations              No
   Record selection (Logic)   No       No        '7 File reading specs      Yes
   Sort order specification   No       No        '8 Level break specs       No

Attention:      Touch FN'0-'8 listed above, EXECUTE to accept, FN'31 to cancel ■
```

Figure 10-3. Batch Generator Function Selection Screen

The Batch Generator Function Selection screen is identical to the Report Utility Selection screen, Figure 5-2, except that there is no reference to line formats, line length, or a printer address. This screen is explained in Section 5.2.

With the exception of FN 05, Field specification, all components to which this screen provides access are identical to those to which the Report Utility Selection screen provides access. Refer to Chapter 6 for instructions about operations, level break specifications, field description, record selection, and sort order specification. Note that, as in reports, you can disable the CANCEL key for the operator during batch processing execution by specifying this as a pass/fail action. Field specification is described in Section 10.4.

10.4 BATCH PROCESSING FIELD SPECIFICATION

When you choose FN 05 on the Batch Generator Function Selection screen (Figure 10-3), the Batch Processing Field Editor screen, Figure 10-4, appears.

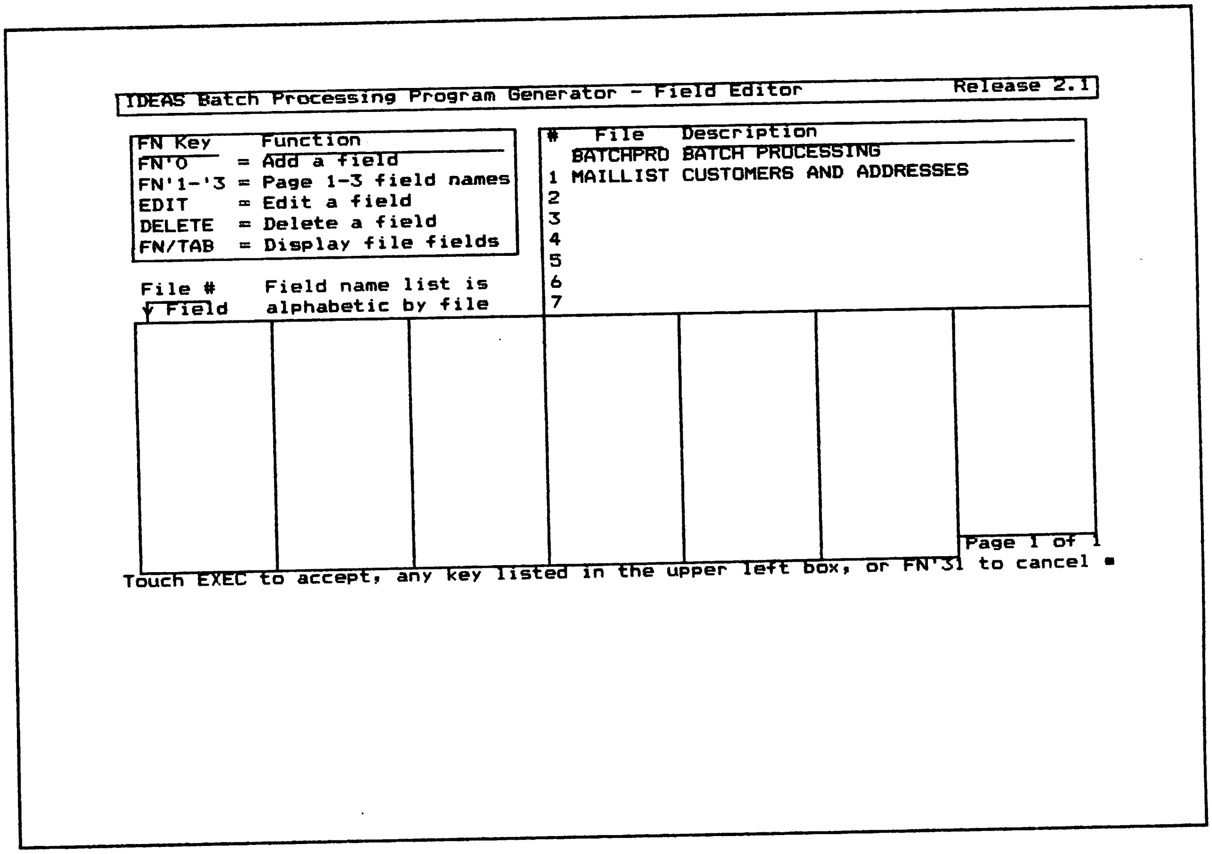


Figure 10-4. Batch Processing Field Editor Screen

This screen appears with a list of all names previously defined for the program. The first time the screen appears, no names are displayed. Up to 83 of the possible 249 fields are displayed. You can move among the three pages of field names by touching FN 01, 02, or 03.

To add a field, touch FN 00. The Batch Processing Field Specification screen, Figure 10-5, appears. To edit or delete a defined field, touch the EDIT or DELETE key; either the first field of the display or the last edited field is highlighted. You can move the highlight itself with the cursor keys (both 2236DE and 2236DW cursor keys are operational) or the Space Bar and BACKSPACE keys. When the field you wish to edit or delete is highlighted, touch EDIT or DELETE once more to complete the operation.

After you touch FN 00 to add a field, the Batch Processing Field Specification screen (Figure 10-5) appears. You define the attributes that appear on this screen in the same way as the attributes on the Report Field Specification screen (Figure 6-5), except that you do not define row, column, floating $, commas, and sign formats on this screen. These are unique to reports.

```
IDEAS Batch Program Generator - Field Specifications                    Release 2.1

  Program    Description                        FN Attribute
  BATCHPRD   BATCH PROCESSING                   00 File number
                                                         name
 File #                                         01 Field name          'FIELD001
 ¥  File      Description
 1 MAILLIST  CUSTOMERS AND ADDRESSES            04 Length                      0
 2                                              05 Position                   80
 3                                              06 Field type                  2
 4                                                    Valid field types:
 5                                                    0 = unsigned numeric
 6                                                    1 = signed numeric
 7                                                    2 = alphanumeric
                                                07 Justify                     N
 Special (reserved) field names:                08 Zero fill                   N
    Name      Description                       09 Decimals                    0
   @SYSMNTH  2-digit month from system date     10 Grouping level
   @SYSDAY   2-digit day   from system date
   @SYSYEAR  2-digit year  from system date     14 Blank after level break     N

Attribute FNs shown above, EXEC=Accept, 16=Field names, 31=Cancel
```

Figure 10-5. Batch Processing Field Specification Screen

The Blank after level break field determines whether the field being defined should be set to blanks (or zeros, for numeric fields) after the operations for the current level are processed.

Unlike the Report Field Specification screen, you cannot use the Batch Processing Field Specification screen to delete a field. You must use the DELETE key in the Field Editor screen (Figure 10-4) to delete a field.

CHAPTER 11
IDEAS SUPPLEMENTARY UTILITIES

11.1 OVERVIEW

The IDEAS Supplementary Utilities are a group of utility programs provided with the IDEAS software that increase the power and versatility of an IDEAS-developed application. When you touch FN 14 at the Primary Program Selection screen (Figure 2-4), the Supplementary Utilities menu, Figure 11-1, appears. Only users of Class F see this menu. A user of any other class who chooses this option sees a menu where the only option is to return to the Primary Program Selection screen. (If you wish to change the security level for a particular utility, you can use the IDEAS menu utilities to revise the menu IDS2MU01.)

```
IDEAS System Utilities - Supplementary Utilities Menu Module          Release 2.1

                      FN Program Module
                    ■ 01 Protect all records
                    · 02 Release all records
                    · 03 File status
                    · 04 Change device addresses
                    · 05 Convert IDEAS2 file to Wang TC file
                    · 06 Convert Wang TC file to IDEAS2 file
                    · 07 Expand size of IDEAS2 file, saving data
                    · 08 Edit file maintenance program
                    · 10 Install an application
                    · 11 Key file recovery
                    · 12 IDEAS1 to IDEAS2 Conversion utilities
                    · 13 Key file record dump utility
                    · 14 CPU system file maintenance
                    · 15 Data file cross-reference utility
                    · 31 Main menu

SEP 27 83                                                          Term #  2
Mem 36.00                   Global Selected                        Sta. # 20

Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 11-1. Supplementary Utilities Menu

The supplementary utilities perform a variety of functions. The individual utilities are described in Sections 11.2 through 11.14.

If you want to run the utilities from an application, you must be sure the IDEAS files necessary for each utility are available to the applications. You can also include the Device Addresses option from the Primary Program Selection menu (Figure 2-4) on a application menu. Appendix F lists the names of the utility programs that you must put on an application menu and the names of the files necessary to run each utility.

When you run the Supplementary Utilities during IDEAS development, any data files you want to operate on are opened automatically. When you run the Supplementary Utilities from an application, however, you must open these data files in the START module for that application.

11.2 PROTECT ALL RECORDS

When you touch FN 01 at the Supplementary Utilities menu, the next screen asks for the file name of the data file that contains the records you want to protect. After you enter a valid file name, a message appears at the bottom of the screen indicating that the records of that data file are being protected.

This utility provides a way to protect all records in a file at the record level by the using partition or CPU. By means of this utility, you can limit access to existing records in a file to one partition, or, in multiplexed configurations, to one partition of a particular CPU.

11.3 RELEASE ALL RECORDS

When you touch FN 02 at the Supplementary Utilities menu, the next screen asks for the file name of the data file that contains the records you want to release. After you enter a valid file name, a message appears at the bottom of the screen indicating that the records of that data file are being released.

This utility is the counterpart of the Protect All Records utility. The utility turns off the record protect byte set by Protect All Records. This allows file access to all records in the file from any calling partition or CPU.

11.4 FILE STATUS

The File Status utility provides data file status information. When you touch FN 03 at the Supplementary Utilities menu, a screen appears that prompts you to enter a file name for the data file you want to review. If the file does not exist or was not opened by the START module, an error message appears. You can enter a different file name or cancel out of the utility. If the data file name is valid, the File Status Report screen, Figure 11-2, appears.

```
IDEAS System Utility - File Status Report                          Release 2.1
==============================================================================

                         File Name: "MAILLIST"

         Number of records specified      500
         Number of records provided       532
         Number of records now used         3
         Number of records available      529

         % Full (as specified)           0.60
         % Full (actual)                 0.56

         Number of overflow records         0
         Overflow percentage             0.00

         Maximum possible number of records/bucket      532
         Maximum actual current records/bucket            3
         Minimum actual current records/bucket            3
         Average actual current records/bucket            3


Attention:      Touch EXECUTE to return to menu, EDIT to enter new file name.  ■
```

Figure 11-2. File Status Report Screen

An explanation of the information that appears on this screen follows.

NUMBER OF RECORDS SPECIFIED
The number of records specified for the data file in the Disk Space Allocation screen (Figure 3-6) of the Data File utilities.

NUMBER OF RECORDS PROVIDED
The number of records specified + n% distributed free space (nominally, n = 5 %; however, this may be different, depending on key length), as indicated in the Disk Space Allocation screen of the Data File utilities.

NUMBER OF RECORDS NOW USED
The number of data records in the file.

NUMBER OF RECORDS AVAILABLE
The number of records that you can add to the file before 100% Full (actual) occurs.

% FULL (AS SPECIFIED)
The percentage full, based on the number of records you specified during file creation. Since IDEAS allocates more records than you specify, this value can exceed 100% in some situations.

% FULL (ACTUAL)
The percentage full, based on the actual number of records provided for during file creation.

Data files in IDEAS systems are divided into sections known as buckets. The file access method that IDEAS uses determines, by a process called hashing, in which bucket a record belongs. (Refer to Appendix D for complete information on the structure of IDEAS data files.) The following fields provide information about bucket status:

NUMBER OF OVERFLOW RECORDS
The number of records that have overflowed from one bucket to another.

OVERFLOW PERCENTAGE
The number of overflow records divided by the number of records now used multiplied by 100

MAXIMUM POSSIBLE NUMBER OF RECORDS/BUCKET
The number of records provided divided by the number of buckets.

MAXIMUM ACTUAL CURRENT RECORDS/BUCKET
The largest number of records contained in any one bucket of the file.

MINIMUM ACTUAL CURRENT RECORDS/BUCKET
The smallest number of records contained in any one bucket of the file.

AVERAGE ACTUAL CURRENT RECORDS/BUCKET
$b_1+b_2+b_3+...b_n$ divided by the number of buckets in the file, where b_x = the number of records in the xth bucket).

11.5 CHANGE DEVICE ADDRESSES

This utility allows you to change the associated device addresses of an IDEAS application without entering the START program revision utility and without manually changing the addresses for each individual data file. When you touch FN 04 at the Supplementary Utilities menu, the Change Device Addresses screen, Figure 11-3, appears.

```
IDEAS Supplementary Utilities -- Change Device Addresses              Release 2.1

'START' module name                                         ■■■■■■■■
Address where 'START' module now resides                    ■■■

                   Peripheral device address assignments
      Old  New                                  #08/ ■■■  ■■■  Application data files
#01/  ■■■  ■■■  Printer                         #09/ ■■■  ■■■  Application data files
#02/  ■■■  ■■■  IDEAS system utilities          #10/ ■■■  ■■■  Application data files
#03/  ■■■  ■■■  Screen & Report masks           #11/ ■■■  ■■■  Application data files
#04/  ■■■  ■■■  Application programs            #12/ ■■■  ■■■  Application data files
#05/  ■■■  ■■■  Data file control files         #13/ ■■■  ■■■  Application data files
#06/  ■■■  ■■■  Telecommunications board        #14/ ■■■  ■■■  Application data files
#07/  ■■■  ■■■  Application data files          #15/ ■■■  ■■■  Application data files

     Do you want to update the control file record of addresses for all
       data files (and any extra volumes) that have been (or will be)
         moved from one platter to another?                     ■ (Y/N)
             Address where control files now reside      ■■■
               Data file addresses:   Old  ■■■    New  ■■■
```

Figure 11-3. Change Device Addresses Screen

You must first indicate the name and current disk address for the START module whose device addresses you want to change. Then, if necessary, enter the new address for each component of the application. This changes the device addresses associated with the device numbers in the START module you entered.

The addresses of the current IDEAS system utilities and START module you entered are the defaults in the new address assignment column. On some occasions, however, you may prepare an application for use at another location with a different configuration. You can specify the address assignments for that system in advance, and they are saved in the START module.

When assigning device addresses for a START module, you must allocate at least one device number for each address that will have data files (or their extra volumes) that application uses. Up to nine different data file addresses can exist. Changing the address at a particular device number to a new address has no consequences except to make the new address available and the old one unavailable (unless it is available as another device number); it does not move files or change information that data control files contain.

You must also indicate whether you want to change device addresses in the data file control files automatically (the actual address for each data file is recorded in its corresponding control file). If you indicate no, you can change selected files manually with the data file utilities. If you indicate yes, you must indicate the current address of the data file control files. The old list of possible addresses is presented to you one at a time. You can now update the control file information on disk addresses for all the data files that were at each old address. All addresses change to whatever new address you enter or accept. You cannot change them to more than one address, except by editing them separately in the data file utilities.

You can keep files at different addresses by entering them as such, or you can consolidate files at different addresses to one address. To do this, however, you must be aware of the locations of any extra volumes. A consolidation that points two volumes of one file to a single address (though they could never be moved to the same address) causes an error message, and you must then correct the multivolume disk address assignment through the data file utilities.

11.6 CONVERT IDEAS FILE TO WANG TC FILE

This utility allows you to convert IDEAS files into Wang standard telecommunications format for data transmission. You can transmit files converted from IDEAS data file format to Wang telecommunications file format to host or remote sites using any of the Wang telecommunications emulators.

You can use the Convert IDEAS File to Wang TC File utility and the Convert Wang TC File to IDEAS File utility for more than TC transmission. For example, if you want to delete all records from an IDEAS file that were entered before a given date, you can transfer only the records entered after that date to a TC file, then initialize the IDEAS file, then transfer the records back from the TC file.

When you touch FN 05 at the Supplementary Utilities menu, the Convert IDEAS2 Data File to TC Format File screen, Figure 11-4, appears.

```
                CONVERT IDEAS2 DATA FILE TO STANDARD WANG TC FILE
IDEAS2 data file name  ********                              TC file name
Should records be concatenated?   (1 = YES, 2 = No)          See note 1

TC file disk address          # of sectors for TC file       See note 3
FN'0:  Minimum key to process                Maximum key to process

********  MOVE ONLY RECORDS WHICH MEET THE FOLLOWING REQUIREMENTS:  ********
  FIELD    OP CONSTANT     (Legal operations are = , < , > , <= , >= , and <> )
1
                                           Use AND or OR
2
                                           Use And or OR
3

Note 1: "Concatenate" means there will be no extra space between the records.
Note 2: "Non-concatenate" is for TC files used outside the IDEAS context, and
        creates a standard Wang format TC file with the entered segment length
        recorded in it. See the TC section of the user manual if necessary.

Note 3: The default is the number of sectors the file will require unless you
        intend to process only some records. You may increase the default.
```

Figure 11-4. Convert IDEAS2 Data File to TC Format File Screen

On this screen, you must enter an IDEAS Release 2 data file name and an output telecommunications file name. Next, indicate whether to concatenate output records. A response of 1 indicates yes, 2 indicates no. Indicating no concatenation yields a standard Wang format TC file; concatenation does not.

Concatenated records are not padded. Concatenated records are written to 80-character TC segments with one record starting immediately at the end of another record. For example, with a 65-character input record, the first record of the input file is written to the first 65 bytes of the output TC segment and the second input record is written to the last 15 bytes of the first output TC segment and first 50 bytes of the second TC segment.

You must also indicate an output address for the telecommunications file. If you specify an invalid or unattached device, an error message appears and you must reenter a valid device selection. If the output file name you entered currently exists on the output device you selected, a warning message appears.

In the next field, indicate the number of sectors to reserve for the output file. The default is the number of sectors the file requires to process all records. You can increase the default. If the telecommunications file already exists on the output device you selected, the number of sectors defaults to the number selected when the file was created. The new file overwrites the previously catalogued file.

You can indicate up to three logical tests for the records to be processed. After you enter the field to be tested and the logical operator (AND or OR), you can enter a value against which to test the field within each record. This value must match the format of the field to be tested, so that numeric fields are compared to numeric constants and alpha fields to alpha constants. Note that given three processing conditions A, B, and C, the statement "process only if condition A AND B OR C" means "process only if conditions both (A AND B) together, OR condition C." Likewise, "A OR B AND C" means "(A OR B) AND C."

Accept the screen to create the telecommunications file. The IDEAS file remains unchanged.

NOTE

If you do not release all records before converting to telecommunications format, another partition cannot convert them back to IDEAS format.

11.7 CONVERT WANG TC FILE TO IDEAS FILE

The Convert Wang Telecommunications file to IDEAS2 data file format is the counterpart of the Convert IDEAS2 data file to Wang Telecommunications file format utility. It converts files previously converted from IDEAS2 files to telecommunications format back to IDEAS file format.

When you touch FN 06 at the Supplementary Utilities menu, the Convert TC File to IDEAS2 Data File screen, Figure 11-5, appears.

```
                    CONVERT TC FILE TO IDEAS2 DATA FILE

 IDEAS2 file name  ■■■■■■■■

 TC file name                                    TC file disk address

 Are records in the TC file concatenated? 1 = YES, 2 = NO (See note)

 If a record to be inserted already exists in the IDEAS2 file:
             1 = Overwrite such records with data from TC file
 Pick one:   2 = Stop for operator choice at each record
             3 = Overwrite no such records

 Do you want to be informed about such records?
             1 = Print (75 bytes of) such records on the printer
 Pick one:   2 = Display (75 bytes of) such records on the screen
             3 = Don't print or display anything

 Note:   "1" means there is no extra space between records in the TC file.
         "2" means the TC file consists of segments, the length of which was
             chosen by the person who created the TC file.
```

Figure 11-5. Convert TC File to IDEAS2 Data File Screen

Enter the name of the IDEAS2 file; this file must already exist. Then enter the name of the telecommunications file you want to convert and the disk address of the telecommunications file.

Indicate whether the records in the telecommunications file are concatenated, as described in Section 11.6. You must indicate the correct option here. If you make the wrong selection, the output data file may be damaged, or incorrectly formatted records may be written into the output file.

You can choose one of three options to be executed if a record already exists in the IDEAS file. These options are: overwrite existing records with data from the telecommunications file; stop execution until the operator chooses whether to overwrite the record; or process the entire file but do not overwrite existing records.

When a record already exists in the IDEAS file, you also have three display options to help you identify the file. These options are: print the records on the printer; display the records on the screen; or do not print or display the records at all. Only the first 75 bytes of the record are listed.

Accept the screen to convert the TC file. The TC file remains unchanged.

11.8 EXPAND SIZE OF IDEAS FILE, SAVING DATA

This utility enables you to expand the size of an IDEAS file without destroying the data the file already contains. The utility involves three steps. The first step converts the data from the IDEAS file to telecommunications format and saves it in a telecommunications file. Second, the Disk Space Allocation screen of the data file utilities appears; you can reestablish the size of the data file. The third step moves the data back from the TC file into the expanded data file.

When you touch FN 07 at the Supplementary Utilities menu, the first screen of the utility appears. This screen is identical to Figure 11-4, the Convert IDEAS Data File to TC Format File screen. For instructions, refer to Section 11.6.

The next screen that appears is the Disk Space Allocation screen, which is the same screen as Figure 3-6. This screen is part of the data file utilities; an explanation of it is given in Section 3.2.

After you modify the data file , a screen appears that allows you to move the data of the telecommunications file you just established back into the IDEAS data file. This screen is identical to Figure 11-5; Section 11.7 contains instructions.

When you use this utility, IDEAS scratches the original data file and renames it. The TC file remains on the disk on which it was defined; you can use this utility to create a backup file. If an appropriate TC file already exists, you can use the existing TC file if it is large enough to contain all the data; if you do this, however, the data currently in the file is overwritten.

You need not open the IDEAS data files to be expanded in the START programs for these utilities.

You can cancel out of these utilities safely from any screen except the Disk Space Allocation screen, where CANCEL is not allowed. If during file expansion, you cancel from the Convert TC File to IDEAS2 Data File screen (Figure 11-5), the IDEAS file is left improperly open and empty. In this case, you must run a START program that opens the file before you can convert a TC file to it.

11.9 EDIT FILE MAINTENANCE PROGRAM

The Edit File Maintenance Program utility provides direct access to the Edit Specification file, allowing you to examine the edits on file and delete unwanted Edit Specifications. Edit Specifications are described in Chapter 5. Only programmers familiar with the structure of the IDEAS system should use this utility; it is not designed for the casual or beginning user.

You cannot enter or modify Edit Specifications with the Edit File Maintenance Program. You can only examine and delete the current Edit Specifications. Exercise caution when deleting edits to be sure that you do not delete actively-used edits and cause problems in program generation.

When you touch FN 08 at the Supplementary Utilities menu, the Edit Specification File Maintenance screen, Figure 11-6, appears.

```
IDEAS Supplementary Utilities  - EDIT SPECIFICATION FILE MAINT.    Release 2.1

 NEW KEY: Screen/Report -- MAILSCRN Field/Operation -- CUST #   Edit# -- 01

        SF'00 - enter new key        SF'11 - next screen of edits
        SF'07 - restart display      SF'12 - next single edit
        SF'09 - delete an entry      SF'15 - expand an edit

    Screen/Report -- MAILSCRN     Field/Operation -- CUST #       Edit# -- 01
        Edit Type -- No-op - Perform Pass/Fails Only

    Screen/Report -- MAILSCRN     Field/Operation -- CUST #       Edit# -- 33
        Edit Type -- Read a record from a data file

    Screen/Report -- MAILSCRN     Field/Operation -- ENDFIELD     Edit# -- 01
        Edit Type -- No-op - Perform Pass/Fails Only

    Screen/Report -- MAILSCRN     Field/Operation -- ENDFIELD     Edit# -- 33
        Edit Type -- No-op - Perform Pass/Fails Only

Attention:      Please Select Appropriate SF Key
```

Figure 11-6. Edit Specification File Maintenance Screen

To indicate the Edit Specification that you wish to examine, enter the name of the screen or report on which you defined the Edit Specification. Next, enter the name of the field or operation associated with the Edit Specification, then the sequence number of the edit. The utility sequentially displays the screen/report name, field/operation name, edit number, and edit type of the Edit Specifications in groups of four. If you leave the Screen/Report name blank, the display starts with the first edit in the file. If you leave the Field/Operation name blank, the display starts with the first edit for the screen or report. If you leave the sequence number blank, the display starts with Sequence 1.

After the Edit Specifications are displayed, you can perform any of the following by touching the indicated special function key:

FN 00 -- ENTER NEW KEY
This option allows you to view a new Edit Specification or series of Edit Specifications. After touching this function key, enter the appropriate fields, just as you did when this screen first appeared.

FN 07 -- RESTART DISPLAY
This option starts displaying the Edit Specifications with the first edit on file.

FN 09 -- DELETE AN ENTRY
This option allows you to delete an entry from the Edit Specification file. It is often easier to delete several specifications with this option than to enter the Screen utilities. You may need to use this option in some circumstances, such as when an edit exists for a screen that is no longer on the system. You can use the cursor keys (FN 05, 06, 11, 12, and 13) to indicate which entry you want to delete.

FN 11 -- NEXT SCREEN OF EDITS
This option displays the next four edits on file.

FN 12 -- NEXT SINGLE EDIT
This option scrolls the display forward one edit.

FN 15 -- EXPAND AN EDIT
This option allows you to look at a particular Edit Specification in greater detail. After you touch this key, the screen changes to allow you to indicate which edit you want to examine. When you indicate the edit, the Edit Specification screen for the edit appears (Figures 5-3, 5-4, 5-5, 5-6, 5-7, 5-8, 5-9, or 5-10). You can either return to the Edit Maintenance utility or view the Pass/Fail Action Specification screen for the edit (Figure 5-11).

11.10 INSTALL AN APPLICATION

This utility copies, as a group, all IDEAS-generated files that meet specified criteria. The files to be copied can be: files contained within an Integrated Support System (ISS) Reference File (ISS is a Wang programming support software package, Wang Package Number 195-0052-3); files whose names fall within a certain range; and files of a particular Version, Application, and/or Function. The utility can also copy the IDEAS files you need to run the application (in case you are installing onto diskette and intend to install from those diskettes), and can copy the Install an Application utility itself.

If any data files contain data to be moved, including the system security file IDS2f005, and if these files must be increased or decreased in size, you must create a TC backup of these files before running the utility. Use the Convert IDEAS File to Wang TC File described in Section 11.6. When you run the Install an Application utility, you must allocate the space you need when you install the data file, as described below. After you run the utility, convert the TC files back to IDEAS format using the Convert Wang TC File to IDEAS File Utility, described in Section 11.7.

Running this utility requires more memory than the rest of IDEAS. In local mode, this utility requires 28K. In global mode, this utility requires a 20K foreground partition and the standard 14K background partition.

When you touch FN 10 at the Supplementary Utilities menu, a screen appears that requests the disk address of the IDEAS Release 2 software. After you enter an address, the Application Installation Screen 1 (Figure 11-7) appears.

```
                 INSTALLATION OF AN IDEAS RELEASE TWO APPLICATION

NOTE:  If any data files (including the system security file "IDS2f005")
       have crucial data AND need to be expanded in size then you must make
       TC backups of them before running this utility.

COPY ALL FILES WHICH MEET THE FOLLOWING REQUIREMENTS:          Version     ■
ISS reference file ■■■■■■■■   Beginning of range ■■■■■■■■   Application ■■■
Address of ISS ref file ■■■   End of range       ■■■■■■■■   Function    ■■■■

What should be done if a file already exists on the destination platter?  ■
Pick one:    1 - Replace all such files     3 - Replace no such files
             2 - Stop for verification for each such file

Are you creating diskettes(1), installing from diskettes(2), neither(3)?  ■
Do you want to just copy (C) or verify as well (V)?                       ■
Do you need to copy the IDEAS modules needed to run an application (Y/N)? ■
Do you need to copy the modules needed to run this utility (Y/N)?         ■

NOTE:  If you supply an ISS reference file name all other criteria will be
       ignored. Otherwise, any file within the range (if supplied) which
       also has the proper version, application and function (if supplied)
       will be copied to the appropriate address (see next screen).
```

Figure 11-7. Application Installation Screen 1

The first field is an optional field that allows you to copy all the files contained within a particular ISS Reference File. If you supply the name of a reference file, you must then indicate the disk address of the file. Other selection criteria are ignored.

If you use an ISS reference file, your data file names and report and batch program names (uppercase) should be in it; the lowercase version of the file names, which represent control files, may or may not be in it. However, any lowercase (control) files that you need to copy must be on the same platter as all the files in the reference file.

If you do not specify an ISS Reference File, you can specify a range of file names that you want to copy. If you leave the beginning and end of the range blank, every file name is included, subject to any other constraints you specify elsewhere on the screen.

You can copy only files of an indicated Version, Application, and/or Function. For information on these three fields, refer to Section 2.2. If you specify a range, Version, Application, and/or Function, only files that match <u>all</u> the criteria you specify are copied.

The utility supplies a default upper range limit of HEX(FF) in each character space. If you leave the Version, Application, or Function field blank, any value in that field is considered acceptable for copying. A blank places no restrictions on the field.

In the next area, you can indicate what to do if a file already exists on the destination platter.

The next screen area contains four questions. The first asks whether you are creating diskettes (answer 1) or installing from diskettes (answer 2). In answer to the second, indicate whether you want simply to copy all files, or to copy and verify them. Next, indicate whether you wish to copy the IDEAS modules needed to run an application. This copy is optional, because the IDEAS modules may already exist on the system where you are installing the application. If you indicate Y, the files contained in the ISS Reference File IDS2fRF7 are copied. Appendix F contains list of the files needed to run an application.

For the fourth question, indicate whether you need to copy the modules needed to run this utility. If you are making diskettes, this allows you to install from those diskettes later. If you are installing an application with no intention of reinstalling, you do not need these files. If you enter Y to this question, the 33 files contained in the ISS reference file IDS2fRF8 are copied. The 33 files are:

IDS2fRF8	IDS2fRF7	IDS2MOVA	IDS2PUI3	IDS2PUI4	IDS2PUI5	IDS2PUI6
IDS2PUI7	IDS2PUI8	IDS2PUIX	IDS2PUIa	IDS2sMV1	IDS2sMV2	IDS2sMV3
IDS2PF11	IDS2PF12	IDS2PF14	IDS2sF11	IDS2s007	IDS2P003	IDS2GLOB
IDS2PUM1	IDS2f000	IDS2LOCL	IDS2PU03	IDS2PU76	IDS2PU77	IDs2SUB1
IDS2SUB2	IDS2SUB3	IDS2SUB4	IDS2SUB5	IDS2SUB6		

After you answer these four questions and accept the screen, the Application Installation Screen 2 (Figure 11-8) appears. Only the bottom four lines of the screen differ from the first application installation screen. In these four lines, indicate the current source address of the various components of the IDEAS application, as well as the destination addresses of each of the components.

```
                INSTALLATION OF AN IDEAS RELEASE TWO APPLICATION

NOTE:   If any data files (including the system security file "IDS2f005")
        have crucial data AND need to be expanded in size then you must make
        TC backups of them before running this utility.

COPY ALL FILES WHICH MEET THE FOLLOWING REQUIREMENTS:           Version
ISS reference file              Beginning of range              Application
Address of ISS ref file         End of range        ¢¢¢¢¢¢¢¢    Function

What should be done if a file already exists on the destination platter?  1
Pick one:    1 - Replace all such files     3 - Replace no such files
             2 - Stop for verification for each such file

Are you creating diskettes(1), installing from diskettes(2), neither(3)?  3
Do you want to just copy (C) or verify as well (V)?                       C
Do you need to copy the IDEAS modules needed to run an application (Y/N)? N
Do you need to copy the modules needed to run this utility (Y/N)?         N
                                           SOURCE ADDRESS   DESTINATION ADDRESS
IDEAS2 system software ---------------     D23              ■■■
Screen & report masks ----------------     ■■■              ■■■
START, menu, & interactive programs --     ■■■              ■■■
Data file control files --------------     ■■■              ■■■
```

Figure 11-8. Application Installation Screen 2

After you accept this screen, the file copies begin. If you are copying to or from diskettes and a diskette is finished or becomes full, a prompt asks you to mount another diskette. If, however, ISS reference files exist on more than one diskette, you must run this utility separately for each diskette.

If you are copying to diskettes, system files and data file control files must all be on the same diskette in order to run this utility from one diskette and to allocate space for the data files later. Therefore, whenever a data file control file must be copied, a prompt appears requesting you to mount the diskette designated for system files and data file control files. After any file that must be on the system diskette is copied, a prompt appears indicating that you can mount a different diskette. This ensures that the system diskette does not run out of room. The files needed to run this utility require about 600 sectors, and each data file control file requires 18. If the rest of your application also fits on the system disk (if, for example, you are using a DSDD diskette), you do not need to make a second diskette.

For each data file you copy, you have three options: to reallocate space; to copy the data file with its data; or to copy and initialize the data file. You can also reallocate space for all data files, copy all data files with data, and copy all data files without data (initializing all data files). If you simply copy data files, you are asked for the source and destination addresses; if you allocate space, you should change the destination address on the allocation screen itself.

Screen and report masks are compressed to minimum size, and program files are copied with no extra sectors. All screens, reports, and batch programs are installed without their Edit Specifications unless the Edit Specifications were archived. Archived screens, reports, and batch programs are copied intact; the whole archive is moved.

The installation utility creates the system address and @TSTFLD file (IDS2fs##, where ## is the CPU number), and the system CPU file (IDS2f001). If you are copying the application onto diskettes, these files are put on the diskette designated for system files and data file control files.

After you run the utility, convert any TC backups you made back to IDEAS format. In addition, adjust all device addresses in the START programs and data file control files for the application to install by running the Change Device Addresses utility, described in Section 11.5.

11.11 KEY FILE RECOVERY

You can use the Key File Recovery utility to reconstruct damaged key files or to put information into an alternate key file created after the entry of data into the primary file. Use this utility for files of Types 2 or 3 only. When you touch FN 11 at the Supplementary Utilities menu, the Key File Recovery screen, Figure 11-9, appears.

```
IDEAS Supplementary Utilities -- Key File Recovery                Release 2.1

      Key File Recovery rebuilds the key portion of a data file
    from the data records in the file.  It is recommended that you
    make a back-up of the file before using this utility to prevent
    data loss in case of operator error or power failure.

  Note:  1. All alternate key files (if any) will be reinitialized.
         2. The primary data file index will be reconstructed.
         3. All alternate key files (if any) will be reconstructed.

  Enter name of primary data file to reconstruct        ■■■■■■■■

  This utility can scan the data file to see if any sectors have been
  damaged.  It will insert the proper IDEAS data file control codes so
  so that there will be no halt in the file recovery or any additional
  data loss.  Do you wish to scan for damaged data sectors? (Y/N)
```

Figure 11-9. Key File Recovery Screen

This utility:

1. Initializes any alternate key files.

2. Initializes the primary key file index (without disturbing the data portion of the file).

3. Determines primary key composition, length, field position within the data record, and sort order from information in the primary key file data file definition.

4. Determines any alternate key composition, length, field position within the data record, and sort order from information in the alternate key file data file definition.

5. Reads the existing data records and reconstructs the primary data file index for complete records or deletes partial records.

6. Reconstructs associated alternate key files.

You can use this utility to create the proper pointers in a newly-created alternate file, even if the primary file and other alternates already contain data. If the new alternate key file does not allow duplicate keys, however, and if its key consists of fields for which records in the primary file would form duplicate alternate keys, running Key File Recovery deletes these records from the primary file. Ensure there are no such records in the primary file or use an alternate file type that allows duplicate keys. (You can recover the primary file by removing the new alternate file and running Key File Recovery again on your backup.)

Enter the name of a primary file; you cannot use an alternate key file for reconstruction. Alternate key files contain only keys and pointers to the data records in the primary data file.

After you enter the name of a primary file, indicate whether to scan the file for damaged data sectors and insert the proper control codes in them. If you bypass this option, a damaged sector causes a 2200 system error D88. You can run the utility again and recover any records that were recoverable when you started. If actual data records are damaged, Key File Recovery reconstructs them when keys can be formed according to specification. Otherwise, the data record is deleted. You should check for yourself whether this process resaves any incomplete or inappropriate information in a record with a valid key.

A second utility screen appears. It contains the names of the primary key file and its associated alternate key files (if any exist) that are to be reconstructed.

If the primary file you are recovering has an associated Type 5 alternate file with fewer records than the primary file, a warning screen appears before the keys are initialized. This screen indicates the number of records in both the primary and the alternate file. You can cancel out of the utility at this point.

After the keys are initialized and recovery begins, a warning message appears when the system encounters the first record to be deleted by a duplicate key. You can cancel out of the utility at this point. If you continue, you can print out the records that are deleted. The audio alarm sounds and a message appears at each deletion.

Keep backup copies of data file control files and primary data files. You can create and maintain backup copies of the data contained in primary data files using the Convert IDEAS File to Wang TC File, described in Section 11.6. You can restore the data records in the BACKUP copies to on-line files using the Convert Wang TC File to IDEAS File utility, described in Section 11.7.

11.12 IDEAS 1 TO IDEAS 2 CONVERSION UTILITIES

These utilities allow you to convert an application developed using IDEAS Release 1 to an application that can be run and modified using IDEAS Release 2. Before you run these utilities, change the device addresses of the START module for this program to conform to your system. To modify the START module, touch FN 04 at the Supplementary Utilities menu. This evokes the Change Device Addresses utility. Follow the instructions given in Section 11.5 to modify the device addresses of the START module IDS2PCST to make it compatible with your system configuration. Running this utility requires 38K of user memory.

When you touch FN 12 at the Supplementary Utilities menu, a screen appears that requests the address where the IDEAS system utilities reside. After you enter this, a screen appears that requests your user ID and password. After you complete these fields, a screen appears that requests the device addresses for the components of the IDEAS Release 1 system you are converting and the device addresses for the new IDEAS Release 2 system. Enter the appropriate addresses. The IDEAS Conversion Utility menu, Figure 11-10, appears.

```
IDEAS2 Conversion Utilites Menu                                  Release 2.1


                         FN Program Module
                       ■ 00 Data File Conversion
                       · 01 Screen Conversion
                       · 02 Report Conversion
                       · 03 Menu Conversion
                       · 04 START Program Conversion
                       · 05 Change Device Addresses
                       · 06 Supplementary Utilities Menu
                       · 31 IDEAS2 Development System


SEP 27 83                                                          Term #  2
Mem 36.00                  Global Selected                         Sta. # 20
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Figure 11-10. IDEAS Conversion Utility Menu

You must choose a separate conversion utility for each component of the IDEAS Release 1 application.

To convert a data file, touch FN 00. A screen appears that requests the name of the data file, the device address of the data file control file, and the device address of the data file itself for both the IDEAS Release 1 file and the new IDEAS Release 2 file. If the file you are converting has associated alternate files, a screen then appears requesting the names and addresses of the converted alternate files. You must change either the file name or the file address when converting an alternate file. The Disk Space Allocation screen (Figure 3-6) appears, first for the primary and then for any alternate files. A file initialization screen then appears. After you initialize the file(s), the records move to the new address in IDEAS Release 2 format.

To convert screen files, report files, menu files, and START programs, press the appropriate function key from the Conversion Utilities menu and follow the prompts on the screen to enter the old and new file addresses.

11.13 RECORD DUMP PROGRAM FOR IDEAS FILES

The Record Dump utility allows you to look at existing IDEAS data files. Appendix D explains the structure of IDEAS data files. Familiarize yourself with this information before using this utility. Only the advanced user who is familiar with the structure of IDEAS files should use this utility.

When you touch FN 13 at the Supplementary Utilities Menu, the Record Dump screen, Figure 11-11, appears.

```
IDEAS System Utility - Data File Record Dump Utility              Release 2.1

Data file name MAILLIST              Description  CUSTOMERS AND ADDRESSES

Bucket #          1 of     1 | 05 Next KIE group | 10 Display Right | 13 Last Block
Index block       1 of     2 | 06 Last KIE group | 11 Next Bucket   | 14 Last Bucket
# KIE's this Bckt          3 | 16 Printer dump   | 12 Next Block    | 15 Display Left

KIE # Byte Pointer V Record#|Key    Record
----- ---- ------- - -------|===== ----- ----1---------2---------3 ---------4--
(C) 1   10    0186 1     390|(available KIE element)
----- ---- ------- - -------|
(C) 1   10    0186 1     390|(available KIE element)
(C) 2   15    0213 1     531|(available KIE element)
    1   20    0189 1     393|....1 ....1 STEVEN.J.LEVINE.......... 30.CAMP.STRE
    2   25    0188 1     392|....2 ....2 KATE.CHAMBERS............ PAUL.REVERE.
    3   30    0187 1     391|....3 ....3 MARLENE.LEVINE........... 11.CARNEGIE.
    4   35    0000 1       0|(available KIE element)
    5   40    0001 1       1|(available KIE element)
    6   45    0002 1       2|(available KIE element)
    7   50    0003 1       3|(available KIE element)
    8   55    0004 1       4|(available KIE element)
    9   60    0005 1       5|(available KIE element)
   10   65    0006 1       6|(available KIE element)
```

Figure 11-11. Record Dump Screen

First, enter the name of the Data File to dump. The parameters of the data file then appear on the screen; these include the current block and index sector being examined, as well as the number of KIEs on record for this bucket. The following information appears on the left side of the screen:

KIE #

This number indicates the KIE number within the current index block. Any coarse index entries are marked with a C. One coarse entry exists for each index block within the current bucket. The KIE listed is that found at the last physical position within the bucket (if any).

BYTE

This number is the byte position within the index block for the KIE entry.

POINTER

This hexadecimal number is the actual pointer used to retrieve records from the file.

V (VOLUME)

This number indicates the volume number in which the record is found.

RECORD #

This number represents the physical location of the file within the data area.

The right side of the screen displays the key and record that were found in the index block and pointer position, respectively. These are displayed in 50-byte segments, with a 10-byte overlap as you move from left to right when examining the data. The appropriate headings and the use of a single dash ruler above the key and a double dash ruler above the record differentiate between keys and records. The rulers are marked in 10-byte increments (1 = Byte 10, 2 = Byte 20). Spaces separate rulers and data fields at points where new primary fields begin. A primary field is any field that is not a subfield of another field. Any record bytes that contain characters less than HEX(10) (printer/display control codes) are translated to HEX(40),@.

The first entry on any screen is the last KIE space within the current index block. If the entry is listed as available, the index block has space available for additional entries. Twelve additional entries that are retrieved from the file in physical order are listed below the separating line.

You can manipulate the information displayed on the screen with the following special function keys:

05 -- NEXT KIE GROUP
06 -- LAST KIE GROUP
These keys allow you to move forward and back through the KIE entries for the current index block. Each new screen shows 12 KIE entries. 2 KIEs overlap from the previous screen.

12 -- NEXT BLOCK
13 -- LAST BLOCK
These keys allow you to move from one index block to another. The KIE separated from the 12 sequential entries reflects the status of the last physical KIE entry position. If the coarse entry is other than HEX(FF), the corresponding index block is full; HEX(FF) indicates there is still space available in the block.

11 -- NEXT BUCKET
14 -- LAST BUCKET
These keys allow you to move from one bucket to another. Each use of these keys resets the current index block indicator and positions the display at the first index block of the current bucket.

10 -- DISPLAY RIGHT
15 -- DISPLAY LEFT
These keys shift the key/record display area left or right. The key/record display area remains at the current position, even if you page back and forth through the KIE entries.

04 -- END KIE LIST
07 -- BEGIN KIE LIST
These keys take you to the beginning or end of the current index block's KIE entries. Other parameters remain the same.

16 -- PRINTER DUMP
A printer dump produces hard-copy documentation of a file. Touching FN 16 clears the lower portion of the screen and replaces it with the current positions used as default values to control the print request. You can change the default values by touching the EDIT key and entering the bucket, index block, and record bytes to be printed. Otherwise, only the bucket or index block currently on the screen prints. Enter the starting and ending byte positions of the record bytes you want to print.

When the print request parameters are correct, accept the screen. The program then checks to see that the printer designated on the device table is available. A box on the lower portion of the screen informs you of the print status; it shows the bucket, block, page and KIE being processed. Touching FN 31 during printing terminates the print request and returns you to the screen display with the parameters that were present before you initiated the request.

The printout header identifies the data file being processed, as well as the current data, bucket, index block, and record bytes.

Line 1 of the printout contains KIE number, byte, pointer, V (volume number), and Record number, as explained above. KIE locations that contain no key are available KIE elements. The remaining area of Line 1 is used when a KIE is listed. In this case, a ruled line separates the entire entry. The line is provided for identification of specific byte positions within the record bytes printed.

Line 2 of the printout contains the key as it is listed in the index block being processed.

Line 3 and all subsequent lines for this entry note which bytes are printed as well as the data. Each line of data is a maximum of 65 bytes long. As many lines as necessary are used to contain the requested bytes. Any record bytes that contain characters less than HEX(20) (printer/display control codes) are translated to HEX(40), @.

11.14 CPU SYSTEM FILE MAINTENANCE

IDEAS allows up to six different CPUs to share a single disk drive. The IDEAS utilities contain a CPU master file that contains the current valid CPU ID numbers. The CPU ID is the number the BASIC-2 function PRINT #ID returns; this requires Operating System Release 2.2 or greater. If the CPU has the necessary hardware, the CPU ID is a 5-digit number; otherwise, it is zero. Section 1.5 describes the uses of the CPU Maintenance file.

This utility allows you to change or delete the CPU ID numbers in the file. When you touch FN 14 at the Supplementary Utilities screen, the CPU File Maintenance screen, Figure 11-12, appears, with instructions presented on the screen.

```
IDEAS2 Extra Utilities -- CPU File Maintenance Screen                Release 2.1

                    CPU NUMBER          CPU ID
                        1               46194
                        2               0
                        3               NOT USED
                        4               NOT USED
                        5               NOT USED
                        6               NOT USED

                    Current CPU ID # is  0

              Existing CPU IDs may be changed or deleted. Any CPU
              in the file may be DELETED except the current CPU.
              The CHANGE option allows the user to swap CPU numbers
              with any other CPU in the file. To ADD a new CPU,
              the IDEAS2 utilities must be run from the new CPU.

Attention:        'EDIT' = Change, FN' 09 = Delete, 'EXEC' = End.
```

Figure 11-12. CPU File Maintenance Screen

11.15 DATA FILE CROSS-REFERENCE UTILITY

When you touch FN 15 at the Supplementary Utilities menu, the Datafile Cross-Reference Utility screen, Figure 11-13, appears.

```
IDEAS Supplementary Utility - Datafile Cross-reference Utility      Release 2.1

        Data File     -  ■■■■■■■■

        Item to       -        SCREENS          (Yy/Nn)
        Process       -        REPORTS          (Yy/Nn)
                      -        BATCH PROGRAMS   (Yy/Nn)

        Processing    -             (MINIMUM)
        Range         -             (MAXIMUM)

        Field Range   -             (MINIMUM)
                      -             (MAXIMUM)

        Version       -
        Application   -
        Function      -

        Report Type   -        0 - NO FIELD ATTRIBUTES
                               1 - FIELD ATTRIBUTES
```

Figure 11-13. Datafile Cross-Reference Utility Screen

This utility provides a convenient cross-referenced list of where the fields of a particular data file are used. After you specify the name of a data file and the parameters of the cross-reference, the IDEAS system produces a list of each occurrence of the data file fields.

The first field to fill is the name of the data file. Enter any data file for which a control file exists. The next three fields allow you to indicate whether you are looking for occurrences of the data file fields in screens, reports, and/or batch programs.

The next two fields specify the range of the screen, report, and batch programs to process.

The next two fields specify the range of fields in the data file to process.

The next three fields specify the Version, Application, and Function of the screen, report, and batch programs to process.

Finally, you can indicate whether you just want a listing of the fields or whether you want the cross-reference to appear with the field attributes specified for each listed occurrence of the field. Specify 0 if you do not want field attributes.

APPENDICES

APPENDIX A
IDEAS IN A MULTIPLEXED ENVIRONMENT

IDEAS Release 2 supports up to six multiplexed CPUs running off of one platter. In order to do this, many system elements are based on an IDEAS-assigned station number, instead of partition number or terminal number. This station number (which appears on all menus) is based on the user's partition number and an IDEAS-assigned CPU number between one and six. The CPU number is based on the CPU ID # (the BASIC-2 function # ID), so you must have BASIC Release 2.2 or greater to implement this feature.

When a CPU accesses the IDEAS development system or an IDEAS START module, IDEAS searches the file that contains a list of CPU numbers (IDS2f001), which resides on the IDEAS platter, for a CPU ID that corresponds to the ID of the current CPU. If an entry is found, IDEAS calculates the station number and sets it in a system-wide variable, R0. This number can range from 1 (for CPU #1, Partition #1) to 96 (for CPU #6, Partition 16). If an entry is not found in the CPU file and room exists in the file for a new entry, IDEAS enters the new CPU in the file and calculates the station number. A Maintenance utility available on the Development System Utilities menu allows you to delete old CPUs from the file or change CPU numbers within IDEAS.

The following system features use the station number:

<u>System work files</u> -- IDEAS assigns a work file to each station for revising screens, programs and reports. The file has 186 sectors. It is called IDS2wSxx, where xx is the station number.

<u>System device addresses and @TSTFLDs</u> -- For each CPU, a system file (IDS2fsxx, where xx is the CPU number) exists. This file is created the first time you use each CPU. This file contains the device address selections used by the IDEAS development system for each partition (15 device addresses for each of 16 partitions), and 160 special system work fields (@TSTFLDs). Each partition is allowed 10 @TSTFLDs (@TSTFLD0-@TSTFLD9) of up to 256 characters each. START modules do not clear these fields. You can change them only by explicitly setting them within an application. You cannot access them from multiple CPUs. In addition, system file IDS2fs01 contains a system-wide "next transaction number" that various applications use. The number increases by 1 after each use.

Record Protection -- When you read a record from an IDEAS file for update, IDEAS sets the record protect flag to the number of the station which is updating the record. Until you save the record back to the file, other stations can read the record for inquiry only; they cannot read it for update. This prevents two or more users from updating a record at one time.

APPENDIX B
LIBRARY SCREENS

IDEAS includes a library of simple screens that contain the Edit Specifications for simple add, change, or delete programs. You can use these screens for reference, or you can copy their Edit Specifications onto your own screens. The library screens include:

IDS2sADD -- Add a record to a data file

IDS2sINQ -- Inquiry into data file

IDS2sSEQ -- Find first logical record, find next logical record in data file

IDS2sACS -- Simple add, change or delete (no checking for duplicate keys on key field entry)

IDS2sACD -- Add, change or delete, with checks for duplicate keys on key field entry

These library screens are supplied on an archive diskette. You can retrieve them from the archive diskette with the Extra Screen utilities on the Screen Utilities menu. After you retrieve them, you can copy Edit Specifications onto your screen with the Copy Edit function for each individual edit or for all edits on a field.

All screens have 10 fields (FIELD001 through FIELD010) in a data file called DATAFILE. FIELD001 is the key field for the primary file. FIELD002 is the key field for an alternate data file called ALTFILE. The key field is the first field on the screen. The Edit Specifications that ADD, EDIT, SAVE or DELETE the record are associated with the field LAST at Row 24, Column 80. The individual logic for each screen is as follows:

<u>IDS2sADD</u> -- Add a record to sample data file.

As a pre-entry pass/fail to FIELD001 (the key field for the primary file), a message appears requesting that you enter the key field. As a post-entry process, the program reads a record from the file for inquiry only, using the key you just entered. If the record exists and is available for use, or if the record exists but another partition is using it, an error message appears and requests a new key. After you fill in the remaining data fields, the field LAST appears at Row 24, Column 80, and a message appears along the bottom of the screen that details your options: touching EXECUTE saves the record, touching FN 31 cancels the operation and loads the specified menu program, and touching EDIT allows you to edit the record. You can touch FN 31 at any field.

<u>IDS2sINQ</u> -- Inquiry into sample data file.

As a pre-entry pass/fail to FIELD001 (the key field for the primary file), a message appears requesting that you enter the key field. As a post-entry process, the program reads a record from the file for inquiry only, using the key you just entered. If the record exists and is available for use, the record appears. If another partition is using the record, an error message to that effect appears and requests a new key. If the record does not exist, an error message appears and requests a new key. After the record appears, the field LAST appears at Row 24, Column 80, and a message appears on the bottom of the screen that details your options: touching EXECUTE recalls another record and touching FN 31 cancels the operation and loads the specified menu program. If you select FN 31, the menu program is loaded immediately. You can touch FN 31 at any field.

<u>IDS2sSEQ</u> -- Find first logical record, find next logical records in data file.

As a pre-entry pass/fail to FIELD001 (the key field for the primary file), a message appears requesting that you enter the key field. The program uses the key you enter to find the first logical record in the data file greater than or equal to the desired key. As a post-entry process, the program reads a record for inquiry only from the file using the key you just entered. To read the record, the program uses the Find First Logical read option on the Read a Record Edit Specification. If the record exists, the program sets system flag D to ON, indicating that the next read is a Find Next option. The record then appears, the field LAST appears at Row 24, Column 80, and a message appears on the bottom of the screen that details your options: touching EXECUTE recalls the next logical record, and touching FN 31 cancels the operation and loads the specified menu program. If you touch EXECUTE, the program branches back to FIELD001, which then finds the next logical record. When the program comes to the end of the file, a message appears that requests a new starting record. If you touch FN 31, the previous menu program is loaded immediately. You can touch FN 31 at any field.

IDS2sACS -- Simple add, change or delete screen (does not check for duplicate records at key entry time).

As a pre-entry pass/fail to FIELD001 (the key field for the primary file), a message appears requesting that you enter the key field. As a post-entry process, the program reads a record randomly from the data file for update or delete using the key you just entered. If the record exists and is available for use, the record appears and processing branches to the field LAST at Row 24, Column 80. If the record exists but another partition is using it, an error message appears and requests a new key. If the record does not exist, you can now enter the remaining data fields. The program does not check for an existing alternate key record when you enter FIELD002. After you enter the remaining data fields, the field LAST appears at Row 24, Column 80, and a message appears on the bottom of the screen that details your options: touching EXECUTE saves the record, touching FN 09 deletes the record, touching FN 31 cancels the operation and loads the previous menu program, and touching EDIT allows you to edit the record. You can touch FN 31 at any field.

If you touch EXECUTE (value of 32 for last function key), the program saves the record to the data file, sets the work buffer to all blanks, displays the blank work buffer on the screen (blanks out the field entries), and branches processing back to FIELD001. If you touch FN 09 (delete), the program deletes the record: the program copies blanks (BLANKFLD) into the record buffer for the data file (FILE # 1) and saves the record. This deletes the key and data area for the last record in the data file that was read for update. If you touch EDIT (last function key equals 33), the program sets system flag G to ON (indicating Edit mode), and you can change the record. After you change it, you can save, delete, or edit the record again. If you touch FN 31, and if a record was not read from the data file, the program loads the previous menu program. If you did not read a record for update, you must save the record back to the data file so that your partition no longer protects it. The subroutine module does this.

IDSsSACD -- Add, change or delete screen that allows all fields to be edited and includes checks for existing records when key is edited.

To allow checking for duplicate keys, this screen uses three dummy fields: OLDKEY1; OLDKYALT; and CHECKKEY. When you edit a record, these fields save the original key fields. If you want to change these fields, you should change the appropriate edits also.

As a pre-entry pass/fail to FIELD001 (the key field for the primary file), a message appears requesting that you enter the key field. As a post-entry process, the program reads a record randomly from the data file for update or delete using the key you just entered. If the record exists and is available for use, the record appears and processing branches to the field LAST at Row 24, Column 80. If the record exists but another partition is using it, an error message appears and a message appears at the bottom of the screen requesting a new key. If the record does not exist, you can now enter the remaining data fields.

After you enter the remaining data fields, the field LAST appears at Row 24, Column 80, and a message appears at the bottom of the screen detailing your options: touching EXECUTE saves the record, touching FN 09 deletes the record, touching FN 31 cancels the operation and loads the specified menu program, and touching EDIT allows you to edit the record. As a pre-entry to the last field, the program tests for the screen-level function key trap on FN 31. If the last function key used is FN 31, program loads the previous menu program immediately. You can touch FN 31 at any field.

The first post-entry process on field LAST is a test for valid entry using a Table Look-Up operation. Valid entries for the field LAST are 09, 31, 32, or 33. Any other entry yields an alarm and a request for a new entry. Next, the program performs the appropriate processing for the option you choose. If you touch EXECUTE (value of 32 for last function key), the program saves the record to the data file, sets the work buffer to all blanks, displays the work buffer (blanks out the fields), and branches processing back to FIELD001. If you choose FN 09 (delete), the program deletes the record: the program copies blanks into the record buffer area for the data file (in this case, BLANKFLD is copied to DATAFILE), and saves the record. This deletes the key and data area for the last record in the data file that you read for update. If you touch the EDIT key (last function key equals 33), the program sets system flag G to ON, indicating Edit mode. The program copies the original keys to the fields OLDKEY1 and OLDKYALT branches processing to FIELD001.

If you change the primary key (FIELD001 does not equal OLDKEY1) the program tests whether a record already exists with the specified new key. If the record already exists, an error message appears requesting a new entry. Otherwise, the program accepts the entry. Similarly, if you change an alternate key field (FIELD002 does not equal OLDKYALT), the program tests whether the record exists and does not allow duplicate keys. After you make all changes, you can save, delete, or edit the record again. If you touch FN 31, and if you did not read a record from the data file, the program loads the specified menu program. If you read a record for update, you must save the record back to the data file, so that your partition no longer protects it. The subroutine module does this.

The IDEAS screen documentation for the library screens is presented on the following pages.

```
Revision number      5        Version             1        User class          0
Last revised 11/11/11         Application code  IDS        User ID code
Last revised by    TJB        Function code    LIBR        Password

         1         2         3         4         5         6         7         8
 12345678901234567980123456789012345678901234567890123456789012345678901234567890
01                                                                              01
02                    SAMPLE SCREEN TO ADD A RECORD TO FILE # 1                 02
03              KEY IS 'FIELD001'; MESSAGE TO SAVE IS ON FIELD 'LAST'           03
04                       SAVE RECORD IS ON LAST FIELD 'LAST'                    04
05FIELD001: ##########          ##########                                      05
06                                                                              06
07FIELD002: ##########                                                          07
08                                                                              08
09FIELD003: ##########                                                          09
10                                                                              10
11FIELD004: ##########                                                          11
12                                                                              12
13FIELD005: ##########                                                          13
14                                                                              14
15FIELD006: ##########                                                          15
16                                                                              16
17FIELD007: ##########                                                          17
18                                                                              18
19FIELD008: ##########                                                          19
20                                                                              20
21FIELD009: ##########                                                          21
22                                                                              22
23FIELD010: ##########                                                          23
24                                                                             #24
 12345678901234567980123456789012345678901234567890123456789012345678901234567890
         1         2         3         4         5         6         7         8
 Data files associated with this screen      Bytes     General screen attributes
1 DATAFILE                                             This is a sub screen       N
                                                       Skip ahead keys enabled    Y
                                                       Skip back keys enabled     Y
                                                       Cancel key (FN'31) enabled Y
                                                       Clear work buffer @ load   Y
                                                       Display all fields @ load  Y
                                                       First line is bright       Y

 Field attribute key:  BEG   Starting Buffer Pos.  KBD   Allow Keyboard Entry
 FLD# Field Number     END   Ending buffer pos.    REQ   Required Entry
 NAME Field Name      *TYPE  Character type (0-9) FULL  Must be Full if Used
 FILE Associated File  JUST  Right justified       EXEC  EXECUTE Required
 ROW  Row on Screen    ZFIL  Zero filled (left)    NBPS  Non-bypassable
 COL  Column on Screen DEC   Decimal places        (by skip ahead/back keys)
 LEN  Length of Field **DISP Display Option (0-3)  #EDT  Number of Special Edits

    *TYPE 0=Digits      2=Digits & .  4=U/C letters  6=U/C,num,punct 8=Yy1=Y,NnO=N
          1=Digits + -  3=Any numeric 5=U/C & digits 7=Any character 9=EXEC,EDIT,FN
   **DISP 0=No display  1=Normal      2=Bright       3=Blinking

 HELP screen name if any = "        " # of fields =  12  Min. record buffer =  256
                                      # of boxes  =   1  Min. work buffer   = 1750
 Trapped function keys (screen level): None
```

FLD #	NAME	FILE (IF ANY)	ROW	COL	LEN	POSITION BEG	POSITION END	TYPE	JUST	ZFIL	DEC	DISP	KBD	RQD	FULL	EXEC	NBPS	# EDT	FLD #
1	FIELD001	DATAFILE	5	11	10	1	10	7	.	.	.	2	Y	Y	.	.	Y	2	1
2	FIELD002	DATAFILE	7	11	10	11	20	7	.	.	.	2	Y	Y	.	.	Y	.	2
3	FIELD003	DATAFILE	9	11	10	21	30	7	.	.	.	2	Y	.	.	.	.	.	3
4	FIELD004	DATAFILE	11	11	10	31	40	7	.	.	.	2	Y	.	.	.	.	.	4
5	FIELD005	DATAFILE	13	11	10	41	50	7	.	.	.	2	Y	.	.	.	.	.	5
6	FIELD006	DATAFILE	15	11	10	51	60	7	.	.	.	2	Y	.	.	.	.	.	6
7	FIELD007	DATAFILE	17	11	10	61	70	7	.	.	.	2	Y	.	.	.	.	.	7
8	FIELD008	DATAFILE	19	11	10	71	80	7	.	.	.	2	Y	.	.	.	.	.	8
9	FIELD009	DATAFILE	21	11	10	81	90	7	.	.	.	2	Y	.	.	.	.	.	9
10	FIELD010	DATAFILE	23	11	10	91	100	7	.	.	.	2	Y	.	.	.	.	.	10
11	DUMMYFLD		5	30	10	103	112	7	.	.	.	0	.	.	.	.	.	.	11
12	LAST		24	80	1	102	102	9	.	.	.	0	Y	.	.	.	.	2	12

Box/Line Documentation:

Type	R	C	V	H
Box	2	12	3	57

Field "FIELD001" pre entry (03) pass/fails only
01 Display message below @ row 24, column 1
"Enter key field for record to be added"

Field "FIELD001" post entry (35) read record
Key consists of: field FIELD001 (Length = 10)
Read from file DATAFILE for inquiry only without system error messages
Read record randomly with key specified above
Set field DUMMYFLD equal to record field FIELD001
If record is found, turn sysflag F ON. If not, turn it OFF
01 If PASS display error message below @ row 24, column 1
"Record already exists-- please re-enter"
02 If PASS branch to field FIELD001
03 If FAIL and sysflag F is ON display error message below @ row 24, column 1
"Record already exists-- please re-enter"
04 If FAIL and sysflag F is ON branch to field FIELD001

Field "LAST " pre entry (04) pass/fails only
01 Display message below @ row 24, column 1
"Key 'EXEC' to save record, 'EDIT' to edit."

Screen "IDS2sADD" - ADD A RECORD TO TEST DATA FILE 09/27/56 Page 2.0001

Field "LAST " post entry (38) pass/fails only
01 If last FN key is EDIT skip ahead 6 P/F's
02 If last FN key is EXECUTE skip ahead 1 P/F
03 Branch to field LAST
04 Display message below @ row 24, column 1
"Saving record to data file"
05 Save record for file DATAFILE
06 Copy all blanks to the work buffer
07 Display from field FIELD001 to last field
08 Branch to field FIELD001

```
Screen "IDS2sINQ" - INQUIRY INTO TEST DATA FILE          09/27/56  Page 1 of 2

Revision number     3        Version             1        User class        0
Last revised 11/11/11        Application code  IDS        User ID code
Last revised by   TJB        Function code    LIBR        Password

         1         2         3         4         5         6         7         8
 12345678901234567980123456789012345678901234567890123456789012345678901234567890
01                                                                              01
02                SAMPLE SCREEN FOR RANDOM INQUIRY INTO FILE # 1                02
03                 KEY IS 'FIELD001'; MESSAGE IS ON FIELD 'LAST'                03
04             LAST FIELD BRANCHES BACK TO KEY AFTER CLEARING SCREEN            04
05FIELD001: ##########                                                          05
06                                                                              06
07FIELD002: ##########                                                          07
08                                                                              08
09FIELD003: ##########                                                          09
10                                                                              10
11FIELD004: ##########                                                          11
12                                                                              12
13FIELD005: ##########                                                          13
14                                                                              14
15FIELD006: ##########                                                          15
16                                                                              16
17FIELD007: ##########                                                          17
18                                                                              18
19FIELD008: ##########                                                          19
20                                                                              20
21FIELD009: ##########                                                          21
22                                                                              22
23FIELD010: ##########                                                          23
24                                                                             #24
 12345678901234567980123456789012345678901234567890123456789012345678901234567890
         1         2         3         4         5         6         7         8
 Data files associated with this screen      Bytes     General screen attributes
1 DATAFILE                                             This is a sub screen      N
                                                       Skip ahead keys enabled   Y
                                                       Skip back keys enabled    Y
                                                       Cancel key (FN'31) enabled Y
                                                       Clear work buffer @ load  Y
                                                       Display all fields @ load N
                                                       First line is bright      Y

 Field attribute key:    BEG  Starting Buffer Pos.   KBD   Allow Keyboard Entry
 FLD# Field Number       END  Ending buffer pos.     REQ   Required Entry
 NAME Field Name        *TYPE Character type (0-9)   FULL  Must be Full if Used
 FILE Associated File    JUST Right justified        EXEC  EXECUTE Required
 ROW  Row on Screen      ZFIL Zero filled (left)     NBPS  Non-bypassable
 COL  Column on Screen   DEC  Decimal places         (by skip ahead/back keys)
 LEN  Length of Field **DISP Display Option (0-3)    #EDT  Number of Special Edits

    *TYPE 0=Digits     2=Digits & .  4=U/C letters  6=U/C,num,punct 8=Yyl=Y,NnO=N
          1=Digits + - 3=Any numeric 5=U/C & digits 7=Any character 9=EXEC,EDIT,FN
   **DISP 0=No display 1=Normal      2=Bright       3=Blinking

 HELP screen name if any = "        "  # of fields =  11  Min. record buffer =  256
                                       # of boxes  =   1  Min. work buffer   = 1750
 Trapped function keys (screen level): None
```

FLD #	NAME	FILE (IF ANY)	ROW	COL	LEN	POSITION BEG	POSITION END	TYPE	JUST	ZFIL	DEC	DISP	KBD	RQD	FULL	EXEC	NBPS	# EDT	FLD #
1	FIELD001	DATAFILE	5	11	10	1	10	7	.	.	.	2	Y	Y	.	.	.	2	1
2	FIELD002	DATAFILE	7	11	10	11	20	7	.	.	.	2	Y	.	.	.	.	.	2
3	FIELD003	DATAFILE	9	11	10	21	30	7	.	.	.	2	Y	.	.	.	.	.	3
4	FIELD004	DATAFILE	11	11	10	31	40	7	.	.	.	2	Y	.	.	.	.	.	4
5	FIELD005	DATAFILE	13	11	10	41	50	7	.	.	.	2	Y	.	.	.	.	.	5
6	FIELD006	DATAFILE	15	11	10	51	60	7	.	.	.	2	Y	.	.	.	.	.	6
7	FIELD007	DATAFILE	17	11	10	61	70	7	.	.	.	2	Y	.	.	.	.	.	7
8	FIELD008	DATAFILE	19	11	10	71	80	7	.	.	.	2	Y	.	.	.	.	.	8
9	FIELD009	DATAFILE	21	11	10	81	90	7	.	.	.	2	Y	.	.	.	.	.	9
10	FIELD010	DATAFILE	23	11	10	91	100	7	.	.	.	2	Y	.	.	.	.	.	10
11	LAST		24	80	1	102	102	9	.	.	.	0	Y	.	.	.	.	2	11

Box/Line Documentation:

Type	R	C	V	H
Box	2	13	3	54

Field "FIELD001" pre entry (04) pass/fails only
01 Display message below @ row 24, column 1
"Enter key field for desired record"

Field "FIELD001" post entry (36) read record
Key consists of: field FIELD001 (Length = 10)
Read from file DATAFILE for inquiry only without system error messages
Read record randomly with key specified above
Set the work buffer area for file DATAFILE equal to the record read
If record is found, turn sysflag D ON. If not, turn it OFF
01 If sysflag D is OFF display error message below @ row 24, column 1
"Record does not exist -- re-enter"
02 If sysflag D is OFF branch to field FIELD001
03 Display from field FIELD001 to last field
04 Branch to field LAST

Field "LAST " pre entry (04) pass/fails only
01 Display message below @ row 24, column 1
"Key 'EXEC' to recall another record, '31 to cancel."

Field "LAST " post entry (38) pass/fails only
01 If last FN key is EXECUTE skip ahead 2 P/F's
02 sound audio alarm
03 Branch to field LAST
04 Copy all blanks to the work buffer
05 Display from field FIELD001 to last field
06 Branch to field FIELD001

```
Screen "IDS2sSEQ" - FIND FIRST/FIND NEXT IN DATAFILE       09/27/56  Page 1 of 2

Revision number     1        Version            1        User class          0
Last revised 11/11/11        Application code  IDS       User ID code
Last revised by   TJB        Function code     LIBR      Password

         1         2         3         4         5         6         7         8
12345678901234567980123456789012345678901234567890123456789012345678901234567890
01                                                                              01
02             SAMPLE SCREEN FOR FIND FIRST/FIND NEXT IN FILE # 1               02
03               KEY IS 'FIELD001'; MESSAGE IS ON FIELD 'LAST'                  03
04           LAST FIELD BRANCHES BACK TO KEY AFTER CLEARING SCREEN              04
05FIELD001: ##########                                                          05
06                                                                              06
07FIELD002: ##########                                                          07
08                                                                              08
09FIELD003: ##########                                                          09
10                                                                              10
11FIELD004: ##########                                                          11
12                                                                              12
13FIELD005: ##########                                                          13
14                                                                              14
15FIELD006: ##########                                                          15
16                                                                              16
17FIELD007: ##########                                                          17
18                                                                              18
19FIELD008: ##########                                                          19
20                                                                              20
21FIELD009: ##########                                                          21
22                                                                              22
23FIELD010: ##########                                                          23
24                                                                             #24
12345678901234567980123456789012345678901234567890123456789012345678901234567890
         1         2         3         4         5         6         7         8
Data files associated with this screen      Bytes    General screen attributes
1 DATAFILE                                            This is a sub screen      N
                                                      Skip ahead keys enabled   Y
                                                      Skip back keys enabled    Y
                                                      Cancel key (FN'31) enabled Y
                                                      Clear work buffer @ load  Y
                                                      Display all fields @ load N
                                                      First line is bright      Y

Field attribute key:   BEG  Starting Buffer Pos.    KBD   Allow Keyboard Entry
FLD# Field Number      END  Ending buffer pos.      REQ   Required Entry
NAME Field Name      *TYPE  Character type (0-9)    FULL  Must be Full if Used
FILE Associated File  JUST  Right justified         EXEC  EXECUTE Required
ROW  Row on Screen    ZFIL  Zero filled (left)      NBPS  Non-bypassable
COL  Column on Screen  DEC  Decimal places          (by skip ahead/back keys)
LEN  Length of Field **DISP Display Option (0-3)    #EDT  Number of Special Edits

    *TYPE 0=Digits     2=Digits & .  4=U/C letters  6=U/C,num,punct 8=Yy1=Y,NnO=N
          1=Digits + - 3=Any numeric 5=U/C & digits 7=Any character 9=EXEC,EDIT,FN
   **DISP 0=No display 1=Normal      2=Bright       3=Blinking

HELP screen name if any = "        "  # of fields =  11  Min. record buffer =  256
                                      # of boxes  =   1  Min. work buffer   = 1750
Trapped function keys (screen level): None
```

FLD #	NAME	FILE (IF ANY)	ROW	COL	LEN	POSITION BEG	POSITION END	TYPE	JUST	ZFIL	DEC	DISP	KBD	RQD	FULL	EXEC	NBPS	# EDT	FLD #
1	FIELD001	DATAFILE	5	11	10	1	10	7	.	.	.	2	Y	Y	.	.	.	2	1
2	FIELD002	DATAFILE	7	11	10	11	20	7	.	.	.	2	Y	.	.	.	.	.	2
3	FIELD003	DATAFILE	9	11	10	21	30	7	.	.	.	2	Y	.	.	.	.	.	3
4	FIELD004	DATAFILE	11	11	10	31	40	7	.	.	.	2	Y	.	.	.	.	.	4
5	FIELD005	DATAFILE	13	11	10	41	50	7	.	.	.	2	Y	.	.	.	.	.	5
6	FIELD006	DATAFILE	15	11	10	51	60	7	.	.	.	2	Y	.	.	.	.	.	6
7	FIELD007	DATAFILE	17	11	10	61	70	7	.	.	.	2	Y	.	.	.	.	.	7
8	FIELD008	DATAFILE	19	11	10	71	80	7	.	.	.	2	Y	.	.	.	.	.	8
9	FIELD009	DATAFILE	21	11	10	81	90	7	.	.	.	2	Y	.	.	.	.	.	9
10	FIELD010	DATAFILE	23	11	10	91	100	7	.	.	.	2	Y	.	.	.	.	.	10
11	LAST		24	80	1	102	102	9	.	.	.	0	Y	.	.	.	.	2	11

Box/Line Documentation:

Type	R	C	V	H
Box	2	13	3	54

Field "FIELD001" pre entry (03) pass/fails only: Do only if flag N is OFF
01 Display message below @ row 24, column 1
"Enter key field for starting record"

Field "FIELD001" post entry (35) read record
Key consists of: field FIELD001 (Length = 10)
Read from file DATAFILE for inquiry only without system error messages
If sysflag D is OFF, read first logical record with key]= that specified above
If sysflag D is ON, read next logical record
Set the work buffer area for file DATAFILE equal to the record read
01 Turn sysflag N ON and turn sysflag D ON
02 If PASS display from field FIELD001 to last field
03 If PASS branch to field LAST
04 If FAIL display message below @ row 24, column 1
"End of file-- please enter new starting position or FN '31"
05 If FAIL copy all blanks to the work buffer
06 If FAIL for field FIELD001 set keyboard avaibility ON
07 If FAIL turn sysflag D OFF
08 If FAIL branch to field FIELD001

Field "LAST " pre entry (05) pass/fails only
01 Display message below @ row 24, column 1
"Key 'EXEC' to view next record, FN '31 to cancel"

Field "LAST " post entry (38) pass/fails only
01 If last FN key is EXECUTE skip ahead 2 P/F's
02 sound audio alarm
03 Branch to field LAST
04 Copy all blanks to the work buffer
05 Display from field FIELD001 to last field
06 For field FIELD001 set keyboard avaibility OFF
07 Branch to field FIELD001

```
Screen "IDS2sACS" - ADD,CHANGE OR DELETE(SIMPLE)          09/27/56  Page 1 of 2

Revision number     5        Version             1        User class          0
Last revised 09/27/56        Application code  IDS        User ID code
Last revised by   SJL        Function code     LIBR       Password

         1         2         3         4         5         6         7         8
 12345678901234567980123456789012345678901234567890123456789012345678901234567890
01                                                                                01
02                 SIMPLE SCREEN TO ADD, CHANGE OR DELETE A RECORD                02
03              KEY IS 'FIELD001'; MESSAGE TO SAVE PRE-ENTRY TO 'LAST'            03
04               SAVE, EDIT OR DELETE RECORD IS ON LAST FIELD 'LAST'              04
05FIELD001: ##########                                                            05
06                                                                                06
07FIELD002: ##########                                                            07
08                                                                                08
09FIELD003: ##########                                                            09
10                                                                                10
11FIELD004: ##########                                                            11
12                                                                                12
13FIELD005: ##########                                                            13
14                                                                                14
15FIELD006: ##########                                                            15
16                                                                                16
17FIELD007: ##########                                                            17
18                                                                                18
19FIELD008: ##########                                                            19
20                                                                                20
21FIELD009: ##########                                                            21
22                                                                                22
23FIELD010: ##########                                                            23
24                                                                               #24
 12345678901234567980123456789012345678901234567890123456789012345678901234567890
         1         2         3         4         5         6         7         8
                                                      General screen attributes
                                                      This is a sub screen       N
                                                      Skip ahead keys enabled    Y
                                                      Skip back keys enabled     Y
                                                      Cancel key (FN'31) enabled Y
                                                      Clear work buffer @ load   Y
                                                      Display all fields @ load  N
                                                      First line is bright       Y

Field attribute key:  BEG   Starting Buffer Pos.  KBD   Allow Keyboard Entry
FLD# Field Number     END   Ending buffer pos.    REQ   Required Entry
NAME Field Name      *TYPE  Character type (0-9)  FULL  Must be Full if Used
FILE Associated File  JUST  Right justified       EXEC  EXECUTE Required
ROW  Row on Screen    ZFIL  Zero filled (left)    NBPS  Non-bypassable
COL  Column on Screen DEC   Decimal places        (by skip ahead/back keys)
LEN  Length of Field **DISP Display Option (0-3)  #EDT  Number of Special Edits

    *TYPE 0=Digits      2=Digits & .  4=U/C letters  6=U/C,num,punct 8=Yyl=Y,NnO=N
          1=Digits + -  3=Any numeric 5=U/C & digits 7=Any character 9=EXEC,EDIT,FN
   **DISP 0=No display  1=Normal      2=Bright       3=Blinking

HELP screen name if any = "          "  # of fields =  11  Min. record buffer =  256
                                        # of boxes  =   1  Min. work buffer   = 1750
Trapped function keys (screen level): None
```

FLD #	NAME	FILE (IF ANY)	ROW	COL	LEN	POSITION BEG	POSITION END	TYPE	JUST	ZFIL	DIEC	DISP	KBD	RQD	FULL	EXEC	NBPS	# EDT	FLD #
1	FIELD001		5	11	10	1	10	7	.	.	.	2	Y	Y	.	.	Y	2	1
2	FIELD002		7	11	10	11	20	7	.	.	.	2	Y	Y	.	.	Y	.	2
3	FIELD003		9	11	10	21	30	7	.	.	.	2	Y	.	.	.	.	.	3
4	FIELD004		11	11	10	31	40	7	.	.	.	2	Y	.	.	.	.	.	4
5	FIELD005		13	11	10	41	50	7	.	.	.	2	Y	.	.	.	.	.	5
6	FIELD006		15	11	10	51	60	7	.	.	.	2	Y	.	.	.	.	.	6
7	FIELD007		17	11	10	61	70	7	.	.	.	2	Y	.	.	.	.	.	7
8	FIELD008		19	11	10	71	80	7	.	.	.	2	Y	.	.	.	.	.	8
9	FIELD009		21	11	10	81	90	7	.	.	.	2	Y	.	.	.	.	.	9
10	FIELD010		23	11	10	91	100	7	.	.	.	2	Y	.	.	.	.	.	10
11	LAST		24	80	1	102	102	9	.	.	.	0	Y	.	.	.	Y	2	11

Box/Line Documentation:

Type	R	C	V	H
Box	2	12	3	57

```
Field "FIELD001" pre entry (03) pass/fails only
01 Display message below @ row 24, column 1
   "Enter key field for desired record"

Field "FIELD001" post entry (35) read record: Do only if flag G is OFF
Key consists of: field    FIELD001                          (Length = 10)
Read from file           for update or delete without system error messages
Read record randomly with key specified above
If record is found, turn sysflag D ON.  If not, turn it OFF
01 If FAIL and sysflag D is ON display error message below @ row 24, column 1
   "Record is in use by another partition"
02 If FAIL and sysflag D is ON branch to field FIELD001
03 If PASS display from field FIELD001 to last field
04 If PASS branch to field LAST

Field "LAST     " pre entry (04) pass/fails only
01 Display message below @ row 24, column 1
   "'EXEC' = Accept, FN '9 = Delete, 'EDIT' = Edit, FN '31 = Cancel"
```



```
Field "LAST     " post entry (39) pass/fails only
01 If last FN key is 09 skip ahead 7 P/F's
02 If last FN key is EXECUTE skip ahead 5 P/F's
03 If last FN key is EDIT skip ahead 2 P/F's
04 sound audio alarm
05 Branch to field LAST
06 Turn sysflag G ON
07 If last FN key is EDIT branch to field FIELD001
08 If last FN key is EXECUTE display message below @ row 24, column 1
   "Saving record to file"
09 If last FN key is 09 display message below @ row 24, column 1
   "Deleting record from file"
10 If last FN key is EXECUTE skip ahead 1 P/F
11 Copy all blanks to nowhere (?)
12 Save record for file
13 Turn sysflag D OFF and turn sysflag G OFF and enable skip back keys
14 Copy all blanks to the work buffer
15 Display from field FIELD001 to last field
16 Branch to field FIELD001
```

Screen "IDS2sACD" - ADD,CHANGE OR DELETE A RECORD 09/27/56 Page 1 of 2

Revision number	1	Version	1	User class	0
Last revised	11/11/11	Application code	IDS	User ID code	
Last revised by	TJB	Function code	LIBR	Password	

```
            1         2         3         4         5         6         7         8
   12345678901234567980123456789012345678901234567890123456789012345678901234567890
01                                                                                  01
02                                    SCREEN TO ADD, CHANGE OR DELETE A RECORD      02
03                                  KEY IS 'FIELD001'; MESSAGE IS ON LAST FIELD     03
04                                 LAST FIELD SAVES OR DELETES RECORD AS DESIRED    04
05FIELD001: ##########          ##########          ##########                     05
06                                                                                  06
07FIELD002: ##########          ##########                                         07
08                                                                                  08
09FIELD003: ##########                                                             09
10                                                                                  10
11FIELD004: ##########                                                             11
12                                                                                  12
13FIELD005: ##########                                                             13
14                                                                                  14
15FIELD006: ##########                                                             15
16                                                                                  16
17FIELD007: ##########                                                             17
18                                                                                  18
19FIELD008: ##########                                                             19
20                                                                                  20
21FIELD009: ##########                                                             21
22                                                                                  22
23FIELD010: ##########                                                             23
24                                                                                 #24
   12345678901234567980123456789012345678901234567890123456789012345678901234567890
            1         2         3         4         5         6         7         8
```

Data files associated with this screen	Bytes	General screen attributes	
1 DATAFILE		This is a sub screen	N
		Skip ahead keys enabled	N
		Skip back keys enabled	Y
		Cancel key (FN'31) enabled	Y
		Clear work buffer @ load	Y
		Display all fields @ load	N
		First line is bright	Y

```
Field attribute key:    BEG   Starting Buffer Pos.    KBD   Allow Keyboard Entry
FLD# Field Number       END   Ending buffer pos.      REQ   Required Entry
NAME Field Name        *TYPE  Character type (0-9)    FULL  Must be Full if Used
FILE Associated File    JUST  Right justified         EXEC  EXECUTE Required
ROW  Row on Screen      ZFIL  Zero filled (left)      NBPS  Non-bypassable
COL  Column on Screen   DEC   Decimal places          (by skip ahead/back keys)
LEN  Length of Field  **DISP  Display Option (0-3)    #EDT  Number of Special Edits

    *TYPE 0=Digits      2=Digits & .   4=U/C letters   6=U/C,num,punct 8=Yy1=Y,Nn0=N
          1=Digits + -  3=Any numeric  5=U/C & digits  7=Any character 9=EXEC,EDIT,FN
   **DISP 0=No display  1=Normal       2=Bright        3=Blinking
```

HELP screen name if any = " " # of fields = 14 Min. record buffer = 256
# of boxes = 1 Min. work buffer = 1750

Trapped function keys (screen level): None

Screen "IDS2sACD" - ADD,CHANGE OR DELETE A RECORD 09/27/56 Page 2 of 2

FLD #	NAME	FILE (IF ANY)	ROW	COL	LEN	POSITION BEG	POSITION END	TYPE	JUST	ZFIL	DEC	DISP	KBD	RQD	FULL	EXEC	NBPS	# EDT	FLD #
1	FIELD001	DATAFILE	5	11	10	1	10	7	.	.	.	2	Y	Y	.	.	.	4	1
2	OLDKEY1		5	31	10	103	112	7	.	.	.	0	.	.	.	.	.	.	2
3	FIELD002	DATAFILE	7	11	10	11	20	7	.	.	.	2	Y	Y	.	.	.	2	3
4	OLDKYALT		7	31	10	113	122	7	.	.	.	0	.	.	.	.	.	.	4
5	FIELD003	DATAFILE	9	11	10	21	30	7	.	.	.	2	Y	.	.	.	.	.	5
6	FIELD004	DATAFILE	11	11	10	31	40	7	.	.	.	2	Y	.	.	.	.	.	6
7	FIELD005	DATAFILE	13	11	10	41	50	7	.	.	.	2	Y	.	.	.	.	.	7
8	FIELD006	DATAFILE	15	11	10	51	60	7	.	.	.	2	Y	.	.	.	.	.	8
9	FIELD007	DATAFILE	17	11	10	61	70	7	.	.	.	2	Y	.	.	.	.	.	9
10	FIELD008	DATAFILE	19	11	10	71	80	7	.	.	.	2	Y	.	.	.	.	.	10
11	FIELD009	DATAFILE	21	11	10	81	90	7	.	.	.	2	Y	.	.	.	.	.	11
12	FIELD010	DATAFILE	23	11	10	91	100	7	.	.	.	2	Y	.	.	.	.	.	12
13	CHECKKEY		5	51	10	123	132	7	.	.	.	0	.	.	.	.	.	.	13
14	LAST		24	80	1	102	102	9	.	.	.	0	Y	.	.	.	.	2	14

Box/Line Documentation:

Type	R	C	V	H
Box	2	16	3	48

Field "FIELD001" pre entry (02) pass/fails only: Do only if flag G is OFF
01 Display message below @ row 24, column 1
"Enter key field for desired record"

Field "FIELD001" post entry (34) read record: Do only if flag G is OFF
Key consists of: field FIELD001 (Length = 10)
Read from file DATAFILE for update or delete without system error messages
Read record randomly with key specified above
Set the work buffer area for file DATAFILE equal to the record read
If record is found, turn sysflag D ON. If not, turn it OFF
01 If PASS display from field FIELD001 to last field
02 If PASS branch to field LAST
03 If FAIL and sysflag D is OFF skip ahead 3 P/F's
04 If FAIL display message below @ row 24, column 1
"Record in use by another partition"
05 If FAIL copy all blanks to the work buffer
06 If FAIL branch to field FIELD001

Field "FIELD001" post entry (38) logical test(s): Do only if flag G is ON
Test 1 passes if field FIELD001 = field OLDKEY1 or field constant blank
01 If FAIL turn sysflag F ON

```
Field "FIELD001" post entry (42) read record: Do only if flag F is ON
Key consists of: field    FIELD001                       (Length = 10)
Read from file DATAFILE for inquiry only without system error messages
Read record randomly with key specified above
Set field CHECKKEY equal to record field FIELD001
If record is found, turn sysflag H ON.  If not, turn it OFF
01 Turn sysflag F OFF
02 If sysflag H is ON display error message below @ row 24, column 1
   "Record already exists-- please re-enter"
03 If sysflag H is ON branch to field FIELD001

Field "FIELD002" post entry (35) logical test(s): Do only if flag G is ON
Test 1 passes if field FIELD002 = field OLDKYALT or field constant blank
01 If FAIL turn sysflag F ON

Field "FIELD002" post entry (37) read record: Do only if flag F is ON
Key consists of: field    FIELD002                       (Length = 10)
Read from file ALTFILE  for inquiry only without system error messages
Read record randomly with key specified above
Set field CHECKKEY equal to record field FIELD002
If record is found, turn sysflag I ON.  If not, turn it OFF
01 Turn sysflag F OFF
02 If sysflag I is ON display error message below @ row 24, column 1
   "Record already exists-- please re-enter"
03 If sysflag I is ON branch to field FIELD002

Field "LAST     " pre entry (05) pass/fails only
01 Display message below @ row 24, column 1
   "'EXEC' = Accept, FN '9 = Delete, FN '33 = Edit, FN '31 = Cancel"

Field "LAST     " post entry (38) pass/fails only
01 If last FN key is 09 skip ahead 10 P/F's
02 If last FN key is EXECUTE skip ahead 8 P/F's
03 If last FN key is EDIT skip ahead 2 P/F's
04 sound audio alarm
05 Branch to field LAST
06 If sysflag G is ON skip ahead 3 P/F's
07 Copy field FIELD001 to field OLDKEY1
08 Copy field FIELD002 to field OLDKYALT
09 Turn sysflag G ON
10 Branch to field FIELD001
11 If last FN key is EXECUTE display message below @ row 24, column 1
   "Saving record to file"
12 If last FN key is 09 display message below @ row 24, column 1
   "Deleting record from file"
13 If last FN key is EXECUTE skip ahead 1 P/F
14 Copy all blanks to record DATAFILE
15 Save record for file DATAFILE
16 Turn sysflag D OFF and turn sysflag G OFF
17 Copy all blanks to the work buffer
18 Display from field FIELD001 to last field
19 Branch to field FIELD001
```

APPENDIX C
USER-CODED SUBROUTINE INTERFACE

Although with IDEAS you can perform almost any programming function most applications require, there are some areas where you may need to create a special hard-coded subroutine or user exit that an IDEAS-generated program can call. You create a subroutine instead of modifying IDEAS-generated code manually, because if you modify IDEAS-generated code, IDEAS may fail to recognize the code as an IDEAS program.

User exits under IDEAS Release 2 must be structured according to specific rules to allow IDEAS to integrate them into a cohesive generated program. These rules are:

1. Each individual subroutine must be a separate program residing on the IDEAS utility platter.

2. Each subroutine must be a marked subroutine (DEFFN'nnn) that does not conflict with the IDEAS subroutine calls. IDEAS subroutines go from DEFFN'32 to DEFFN'79; a user-defined subroutine must be greater than 79.

3. You can pass a minimum of 0 and a maximum of 3 variables (fields) to the subroutine. These can be any combination of alpha and numeric values obtained from fields, @TSTFLD's, or @SYSBUF0.

4. If you pass 1 or 2 alpha values, they can have a maximum length of 256 bytes each. If you pass 3 alpha values, the second has a maximum length of 249 bytes (because of current buffer sizes). The first and last can be 256 bytes. IDEAS generates the subroutine calls using a subset of the following variables, depending on the number and order of the receiving alpha and numeric variables:

X,Y,Q	
E4$()	256 bytes
E0$()	249 bytes
F$()	256 bytes -- F$() can be larger than 256, but a maximum of 256 are passed to the subroutine.

You must use arrays for the alpha variables to provide for the possibility that large fields may be passed. The alpha variables can also be the receiving variables if the order in the DEFFN' does not conflict with the order in the generated GOSUB'. All possible subroutine call combinations are listed below:

No variables passed:

```
GOSUB'XXX
```

One variable passed:

Numeric:
```
GOSUB'XXX(Q)
```

Alpha:
```
GOSUB'XXX(E4$())
```

Two variables passed:

Both numeric:
```
GOSUB'XXX(X,Q)
```

1 alpha, 1 numeric:
```
GOSUB'XXX(X,E4$())
GOSUB'XXX(F$(),Q)
```

Both alpha:
```
GOSUB'XXX(F$(),E4$())
```

Three variables passed:

All numeric:
```
GOSUB'XXX(X,Y,Q)
```

2 numeric, 1 alpha:
```
GOSUB'XXX(X,Y,E4$())
GOSUB'XXX(X,F$(),Q)
GOSUB'XXX(E4$(),Y,Q)
```

1 numeric, 2 alpha:
```
GOSUB'XXX(X,F$(),E4$())
GOSUB'XXX(F$(),Y,E4$())
GOSUB'XXX(F$(),F0$(),Q)
```

All alpha:
```
GOSUB'XXX(F$(),F0$(),E4$())
```

You do not need to use the variables listed above as receivers in the user exit. If you use them, they must be in the order listed for each combination.

5. To avoid conflict with variable usage in IDEAS and other Wang utilities, use x, y, z, and w-based variables, though not x, y, z, and w themselves, which are system scratch variables. In other words, the variables available to the user that cause no conflict are:

   X0 through X9, X0$ through X9$
   Y0 through Y9, Y0$ through Y9$
   Z0 through Z9, Z0$ through Z9$
   W0 through W9, W0$ through W9$

   You can use IDEAS system variables if their use is consistent with IDEAS usage.

6. You must create the user exit program module as follows:

   a. Number lines starting with line 1 and increasing the line number by 1 for each line.

   b. Line 1 must consist of a REM followed immediately by the program name without any quotation marks.

   c. Line 2 must consist of a REM followed immediately by a functional description of the user exit. It can be longer than 64 characters, but only the first 64 appear on the specification screen.

   d. Line 3 must consist of a REM followed immediately by one of the text strings listed below. The asterisks represent a description of the return value (up to 20 characters). Use this both for documentation and to generate the proper subroutine calls.

      NO RETURN VALUE
      ALPHA RETURN VALUE = ********************
      NUMERIC RETURN VALUE = ********************

   e. Line 4 must consist of the DEFFN' statement only

   f. Lines 5, 6, and 7 are optional. If you use them, and they each begin with a REM, the characters immediately following the REM (up to 20) are used as descriptions of the receiving variables on the specification screen and for documentation. The receiving variables are the arguments used in a DEFFN' statement.

g. Lines 1 through 4 can contain nothing else. These lines must be single-statement lines with the REMs described above. Lines 5 through 7 can be single-statement REMs (as in f above), or you can omit them. Program text lines must start with line 8, and can contain more than one statement.

h. If you need a return value, you must code the module so that the return value remains in E4$() if it is an alpha value, or in Q if it is a numeric value.

i. If a test within the user exit provides a pass/fail condition, you must set H = 0 or a negative value to signal a fail condition. A pass condition is the default value that IDEAS sets before the subroutine call.

j. IDEAS sets a pass condition before it branches to the user exit (H=1). If the user exit sets a fail condition (sets H = 0 or a negative value), the return value is not placed in its receiver field and that field remains unchanged. Refer to Figure 5-10.

k. A RETURN statement must end the subroutine.

Example:

```
0001 REM USEREXIT
0002 REM FIND THE HYPOTENUSE OF A RIGHT TRIANGLE, GIVEN BASE &
     HEIGHT
0003 REM NUMERIC RETURN VALUE = THE HYPOTENUSE
0004 DEFFN'255(I,Q)
0005 REM BASE OF TRIANGLE
0006 REM HEIGHT OF TRIANGLE
0007 Q=SQR(I*2+Q*Q)
0008 RETURN
```

APPENDIX D
IDEAS FILE STUCTURE

IDEAS Release 2 supports seven types of keyed data files. Although significant differences exist among these file types and their uses, their actual organization on disk is similar. Every IDEAS file is a standard 2200 BASIC-2 disk catalog data file with an EOD sector following the data and an EOF sector at the end of the file.

Buckets

Each IDEAS file consists of one or more "buckets". A bucket is a small indexed file that consists of a two-level index and, in some cases, a separate data record area. A hashing algorithm assigns each key to a "home bucket". If the bucket is full, the key overflows into the next adjacent bucket. An overflow from the last bucket goes into the first bucket. If the bucket into which the key overflows is also full, another overflow occurs until either an unfilled bucket is found or the file is full. All buckets within a given file are identical in size and organization.

Index

Every IDEAS file has a two-level index in each bucket. The index consists of one or more 8-sector blocks that are saved as 24 83-byte elements, for a total of 1992 bytes. The first 9 bytes of each index block hold control information; Table D-1 summarizes the information contained in these bytes. The remaining bytes hold key index elements (KIEs) or are not used; the structure of key index elements is described below. The first index block in each bucket contains the high-order index for that bucket, in addition to the normal KIEs or low level-index.

Table D-1. The First Nine Bytes of an Index Block

Byte(s)	First Index Block/Bucket	Subsequent Index Blocks (if any)
1	Number of remaining index blocks in the bucket	Number of remaining index blocks in the bucket
2	Number of the next index in the current bucket with KIE available.	Not currently used
3	Starting byte within the index block of the first KIE in the block	Starting byte within the index block of the first KIE (always HEX(0A) -- Byte 10)
4 and 5	Starting byte of the last KIE in the block in binary	Starting byte of the last KIE in the block in binary
6 and 7	Number of records (or KIEs) currently used in the bucket in binary.	Not currently used
8 and 9	Number of records (or KIEs) for which this is home bucket but that have overflowed to other buckets	Not currently used

In the first index block, the high-order index for the bucket starts at Byte 10. This consists of the highest (last) KIE in each index block of the bucket and cannot (by definition) extend beyond Byte 249. These KIEs can be vacant, or they can contain real keys.

The KIEs that comprise the low-order index start at the byte immediately following the high-order index in the first block and at Byte 10 in subsequent blocks (the starting byte is the value of Byte 3). They can extend to Byte 1992, although there are two reasons why they cannot fill the area: since KIEs are not split between index blocks (each index block contains an integral number of KIEs), the length of the KIEs may be such that the last KIE ends at a byte lower than 1992; or, the bucket may not contain sufficient records to use all the potential KIE locations in the index block.

When KIEs are added to an index block, they are always inserted into the list of KIEs in ascending sort order and the high-order index is updated.

Key Index Elements (KIEs)

A KIE consists of a packed key, and, in some cases, a pointer or data record in one of the formats shown below:

File Type	KIE Construction
1	KKKKKKKK
2,3,5,6,7	KKKKKKKKPPp
4	KKKKKKKKSDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD

Where: K = The packed key
P = The pointer to the data record
p = Pointer extension, if necessary
S = Protect byte (protecting station number)
D = Packed data record

Key

The key portion of the KIE consists of the key to the data record (which may be a concatenation of up to 5 fields). The key is packed, and, if it is in descending sort order, complemented.

Any numeric fields at the start of the key are packed 2:1 (only an even number of bytes.) These bytes are packed using a HEXPACK statement, but to preserve sort order, they are first translated as follows:

Initial values	$	+	,	-	.	0	1	2	3	4	5	6	7	8	9
Translated values	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?

Uppercase fields that occur at the start of the key or immediately following the packable numeric fields are packed 4:3 (in groups of 4 bytes). An uppercase field is any field that allows only a subset of hex codes 20 through 5F. These are packed by adding HEX(E0) to change the set to hex codes 00 through 3F (6 bits) and using a series of ROTATE Cs.

Any field that allows lowercase characters, or follows such a field, is not packed.

Pointer

The pointer can be either 2 or 3 bytes in length. File Types 1 and 4 do not contain a pointer.

- For single-volume files that contain less than 65536 records, the pointer is 2 bytes.
- For single-volume files that contain more than 65535 records, and for all multivolume files, the pointer is 3 bytes.

The first 2 bytes of the pointer contain a record number in binary that starts at HEX(0000).

When a third byte is necessary, its low-order 5 bits (10, 08, 04, 02, 01) represent a binary value of the number of times that the number 65536 must be added to the record number (0 through 31). The high-order 3 bits (80, 40, 20) represent the binary value of the offset of the volume number (0 through 7).

Pointer = RRRRRRRR RRRRRRRR (vvvrrrrr)

Where: R = Record number
v = Volume
r = Record number extension

The record number is the actual record number in the volume where a new key assigned to the KIE is stored (a 2-byte definition implies Volume 0.) A multivolume file has as many sets of record numbers (each starting at zero) as there are volumes. With this pointer structure, an IDEAS file (except for Types 1 and 4) can contain a maximum of 2,097,152 records per volume, or a maximum of 16,777,216 total records. Types 1 and 4, because they are not dependent on pointers, are limited only by disk size, merge array size if they are to be processed sequentially, and the fact that they are always single-volume files.

In the case of multivolume files, the records are distributed in the following manner:

1. All index blocks are always on Volume 0.

2. The pointer is 3 bytes in length rather than 2. This causes fewer possible KIEs per index block and changes some of the information in the 9-byte control area.

3. The records are stored in inverse order by block. This means that the first 17 records saved are records 12 to 0 on volume 1, followed by all records on Volume 0 from Record 44 to 0 to complete Block 1. Block 2 is used next, with the first records being those on Volume 2 from Number 24 to 0, followed by the remaining records for Volume 1, Numbers 34 to 13.

4. The distribution and order of record storage depends on key length (hence, number of KIEs per block) and number of records per volume.

Pointers for alternate key files (Types 5, 6, and 7) do not exist when the file is initialized. They are initially set to HEX(FF)s, and take the value from the primary file's pointer when a new key is added to the file. If the key file dump module is used to view the index of one of these files, all unused KIEs exhibit a pointer of HEX(FFFF) or HEX(FFFFFF) (depending on the primary file's pointer length) and a record number of 65535. If the primary file is multivolume, or contains more than 65535 records, unused alternate KIEs also exhibit a volume number of 8 (if the key dump module volumes are numbered 1 through 8.)

Data in the KIE

Only Type 4 files contain data records in the KIE. They do not have pointers, nor do they have separate data blocks. The key portion of the KIE is identical to all other file types, but it is followed immediately by one protect byte and the packed data record. The protect byte is normally set to HEX(FF) (unprotected). When a station protects a record, however, this byte is set to the binary value of that station number.

The data record is packed the way a key is packed. Starting numeric fields are packed 2:1, and uppercase fields starting at the beginning of the record or immediately following starting numeric fields are packed 4:3. The only difference is that a different packing algorithm is used for the 4:3 packing of records. This is because keys must be preserved in sort order. Since records do not have to be preserved in sort order, a much faster packing algorithm is used.

For both keys and records, if the number of numeric bytes to be packed is uneven, the last such byte is considered to be part of the upper case string and only the even number of numeric bytes is packed. Only uppercase strings of lengths that can be divided evenly by 4 are packed. Any remaining bytes are considered to be part the unpackable characters.

Type 1 Files

Type 1 files have no pointers, no protect bytes, and no data segments in the KIE. The KIE, and the file itself, consists only of keys. Because there are no protect bytes, you cannot protect "records" (keys) in a Type 1 file. This file type is used for table look-up and validation.

File Organization Commonality

All file types (1 through 7) have identical organization except for the contents of the KIE.

Sequential Processing

IDEAS requires a "merge array" -- E6$() -- to perform sequential processing operations. The keys within each bucket are maintained in sort order; however, keys are intermixed among the buckets according to the hashing algorithm. Sequential processing is performed by reading 1 KIE from each bucket and putting these KIEs into the merge array in sort order. The merge array must be large enough to support any file that you want to process sequentially. To find the required size for the merge array for any given data file, refer to the data file documentation that the IDEAS data file utilities supply, or multiply the number of buckets in the file plus one by the length of the packed key plus the length of the pointer (if any) plus 4.

Each element in the merge array consists of the packed key and pointer (if any) from a KIE in one of the buckets, plus an additional 4-byte pointer telling IDEAS where it found the KIE (absolute sector address and byte number). There is one more merge array element than there are buckets in the file.

<u>Record Blocks</u>

For Type 2 and Type 3 files, IDEAS automatically arranges records into physical blocks of from 1 to 8 sectors to preserve disk space. All record blocks in a given file are the same size.

Records in a Type 2 or Type 3 file consist of a 1-byte protect flag (discussed in the section on Type 4 file KIEs) followed by the packed record. The maximum physical packed record size is 1991 bytes plus the protect byte. The actual unpacked record length may be larger than 1991 if there are enough characters that can be packed to shrink packed size to 1991 or less.

<u>Distributed Free Space</u>

When you specify a number of records for the file, IDEAS adds a minimum of 5 percent more records. This reduces the number of overflows caused by imperfect hashing. Depending on factors such as key size, record size, and number of records, IDEAS may add more than 5 percent. IDEAS first computes the number of index block sectors and record block sectors necessary to create a file that is 105 percent as large as specified and then recomputes the number of records to be provided so that as many are provided as fit in that file configuration. As a result, with no additional space taken up on disk, extra records are usually available beyond the standard 5 percent.

<u>Keys Duplicated in Record</u>

Key fields appear in the index block and in the record itself. This duplication of data requires little additional space in most cases and provides a key file reconstruction capability in the case of all but Type 1 and Type 4 files.

APPENDIX E
SYSTEM-RESIDENT SUBROUTINES

All IDEAS-based applications make use of the system-resident subroutines. The system-resident subroutines are a set of subroutine calls that perform a variety of functions for both the IDEAS development system and applications. These subroutines are contained in a number of individual programs:

Name of Module in Global Mode	Name of Module in Local Mode	Description of Module
IDS2GLBL	IDS2LOCL	Subroutine loader
IDS2SUB1	IDS2SUB1	Screen and field subroutines
IDS2SUB2	IDS2SUB2	Error message subroutines
IDS2SUB3	IDS2SUB3	Key file and record maintenance
IDS2SUB4	IDS2SUB4	Record PUT
IDS2SUB5	IDS2SUB5	Record GET
IDS2SUB6	IDS2SUB6	Field level security
IDS2SUB7*		Date validation and conversions
IDS2SUB8*		Sequential processing

*Automatically loaded if global partition is of sufficient size.

Dates and sequential processing are never loaded as part of the main subroutine module in the VP or in an MVP partition that does not have access to global subroutines. To preserve memory in the local environment, these routines are loaded automatically only when necessary. If a global partition is not large enough to contain the entire set of modules, when the date processing and sequential processing modules are necessary they are paged into the local partition and treated as in local mode.

Other than the differences in loading date logic and sequential operations, the only difference between the global mode and local mode subroutines is in IDS2GLBL and IDS2LOCL. You must load IDS2GLBL specifically, though IDEAS loads IDS2LOCL automatically. IDS2GLBL defines the partition as global and IDS2LOCL simply branches to Line 1000. No real distinction exists between the actual programs in local mode and global mode; the only difference is in how the programs are loaded.

Table E-1 presents a list of the system-resident subroutines available to IDEAS. A more complete explanation of each of the system-resident subroutines follows the table.

Table E-1. IDEAS Subroutine Modules

Subroutine	Description
<u>Screen</u>	Located in IDS2SUB0
DEFFN'32(F5$)	Get field parameters and display screen
DEFFN'33(F5$)	Get field parameters, do not display screen
<u>Field</u>	Located in IDS2SUB0
DEFFN'34(F)	Data entry from keyboard into field number F
DEFFN'36	Display all fields starting at current field number F
DEFFN'37(Q)	Display the contents of the field number Q
DEFFN'43(Q)	Retrieve contents of field by number
DEFFN'44(E$)	Retrieve contents of field by name
DEFFN'45(Q,E4$())	Put contents of E4$() in field number Q
DEFFN'46(E$,E4$())	Put the contents of E4$() into field E$
DEFFN'54(E$)	Get field number of given field name
DEFFN'55(N,A)	Round and put a numeric field into work buffer
DEFFN'58	Internal field level security and indirect operations
DEFFN'64(T,E)	Copy a field by number
DEFFN'65(F7$,E7$)	Copy a field by name
<u>Screen Message</u>	
DEFFN'35(F0$)	Display error message on Line 24 and sound audio alarm
DEFFN'53(F0$)	Display message with "Attention" on Line 24
DEFFN'63(E8$,F0$,E8$(1))	Display 80-byte message at (24,0)
<u>File Access</u>	
DEFFN'41(F5$,E$,J)	Read a record from file F5$
DEFFN'42(F5$,J)	Write a record into file F5$
DEFFN'47	Get file attributes
DEFFN'50(H,W,E9$)	Read an index sector
DEFFN'51(E4$())	Complement and pack key
DEFFN'52	Key access (part of '59 used by sequential processing)
DEFFN'59(V,J,E$,X)	Key file access subroutine (insert/retrieve/delete)
DEFFN'60(F5$,E$)	Set up sort array
DEFFN'61(F5$,E$,J)	Find first logical record =E$
DEFFN'62(F5$,M,J)	Find next logical record in file F5$
DEFFN'66(V,C)	Unpack record

Table E-1. IDEAS Subroutine Modules (continued)

Subroutine	Description
File Access (cont.)	
DEFFN'67(V,F3$,X)	(Internal) Get record given pointer
DEFFN'69(V)	Form key given record and file number
DEFFN'78(F5$)	Set V = file number of file named by F5$
DEFFN'79	Get a record (same as '41 without getting file number)
Print	
DEFFN'48(C,E4$(),L)	Prepare print buffer
DEFFN'49(L)	Print a line of the print buffer
Date	
DEFFN'56(E8$,A)	Date validation subroutine
DEFFN'57(E8$,A,B)	Convert alpha date in format A to format B
Miscellaneous	
DEFFN'38(O,E8$)	(Internal) System error messages
DEFFN'39(F5$,P)	Get limits of file F5$ on device number P
DEFFN'40(E8$)	Load module E8$

DEFFN'32(F5$) -- Get field parameters and display screen.

Input: F5$ -- Screen name

Returns: E1$ -- Help screen name
F6$(11) -- Cancel
F6$(12) -- Skip-ahead
F6$(13) -- Skip-back
E2$() -- Screen attributes
F0 -- Number of fields in screen

Pass the name of an IDEAS-developed screen mask file to this subroutine. The subroutine loads the screen attributes into buffer E2$() from disk (Device #3, unless the screen name begins with IDS2 or ids2, in which case the screen mask is loaded from disk device #2), and displays the screen. The field parameters associated with the specified screen are loaded into E2$(). The work buffer E$() is cleared only if you specified this during development.

The variable F (field number) is set to zero, and the variable F0 is set to the number of fields defined for the specified screen.

DEFFN'33(F5$) -- Get field parameters for screen F5$, do not display screen.

Input: F5$ -- Screen name

Returns: E1$ -- Help screen name
F6$(11) -- Cancel
F6$(12) -- Skip-ahead
F6$(13) -- Skip-back
E2$() -- Screen attributes
F0 -- Number of fields in screen

Pass the name of an IDEAS-developed screen mask to this subroutine. The system loads the field parameters associated with the specified screen into E2$(). It loads from disk device #3, unless the screen name begins with IDS2 or ids2, in which case it loads from disk device #2.

The variable F (field number) is set to zero. The variable F0 is set to the number of fields defined on the specified screen.

This subroutine does not change the screen display.

DEFFN'34(F) -- Data entry from keyboard into field number F.

Input: F -- Field number

Output: T -- Field type
R -- Row
C -- Column
P -- Position
L -- Length
E$,E4$() -- Contents of field
Q -- Numeric value of field or 0

When you pass a valid field number to this subroutine, the system starts processing the field you specified according to the specifications the field parameter record for that field contains. These are the specifications that you defined when you last created or revised the screen; they are shown in the screen documentation. The actual processing of the field varies significantly depending on the field parameters. One factor that remains constant, however, is that no matter how the subroutine processes the field, the contents of the field after the processing is complete always return to the application program in the variables E$ and E4$(). If the contents of the field represent a valid numeric string, the numeric representation of the field returns in the variable Q; otherwise, Q is set to zero.

Internal Operation of DEFFN'34

When the subroutine is called, the appropriate field parameters are unpacked from the screen field parameters. If there is no field parameter record for the specified field (F > F0), the field number returns with a value of zero. Provided there is a valid defined field, however, the following steps occur:

1. The system positions the cursor at the beginning of the field. If the field is blank and is available for input from the keyboard, a string of boxes, equal in length to the field, appears. If the field is not blank and can be displayed, the contents of the field are displayed. If the field cannot be displayed, and cannot be entered from the keyboard, a string of spaces equal in length to the field length appears.

2. If the field is not available for input from the keyboard, skip to Step 6.

3. The cursor returns to the beginning of the field for keyboard input. Only those characters in the specified allowable character set for the field are permitted. If an invalid character is entered, the system ignores it, the audio alarm sounds, and an error message appears on Line 24 that specifies the valid character set.

Exceptions to Step 3 include:

a. EXECUTE (or RETURN) can be used at any time to terminate a field, except when:

   - The field is a REQUIRED field and is currently blank.

   - You specified the field as FULL IF PRESENT and it contains any blank character.

b. BACKSPACE is valid at any time other than when the cursor is at the beginning of the field.

c. Certain special function keys override Step 3.

   You can enable certain special function keys when you create the screen to allow cancel, skip-ahead, and skip-back. These operations are handled in the subroutine module. You can trap other function keys at the screen level which then pass control to the pre-entry operations of the last numeric field. At the field level, function keys are accepted as input and are handled within post-entry or whichever process you specified. The value is held in E3$8.

   If you touch the EDIT key, the field is underscored and the BEGIN (FN'07), END (FN'04), INSERT (FN-10), DELETE (FN'09), 5RIGHT (FN'11), 1RIGHT (FN'12), 1LEFT (FN'13), and 5LEFT (FN'14) operate until you terminate EDIT mode by touching EXECUTE. This completes the entry of a field that does not require EXECUTE for termination. You can also touch the RECALL key (FN'15) to display the original contents of the field again.

If you enable the CANCEL key, touching it at any time other than in EDIT mode cancels the current program, releases any currently retrieved and protected records in the files, and loads the last MENU (or whichever program you specified in the variable R3$(1)). CANCEL can also be handled as a screen-level function key.

If you enable the SKIP-AHEAD keys:

- FN 04 (END) causes the cursor to skip to the end of the screen.
- FN 11 (5 RIGHT) causes the cursor to skip ahead 5 fields.
- FN 12 causes the cursor to skip to the next field.

Any of these three operations displays the original contents of the field in which the key was touched.

If you enable the SKIP-BACK KEYS:

- FN 07 (BEGIN) causes the cursor to skip back to the first field on the screen.
- FN 13 (5 LEFT) causes the cursor to skip back to the previous field.
- FN 14 causes the cursor to skip back 5 fields.

4. If the field is available for input from the keyboard but cannot be displayed, the system accepts keystrokes and puts the data into the field but displays none of the input data.

5. You can terminate the field (other than as described in Step 3 above) as follows:

   a. Filling the field if the field specifications do not require EXECUTE.

   b. Touch EXECUTE.

   c. Use a hyphen (-) as any but the first character in a numeric field that accepts signs as valid characters. You can use a hyphen as the first character in a numeric field to make the field negative, in which case you must touch EXECUTE to terminate the field or the field must be full. You can also use a hyphen in place of EXECUTE both to terminate the field and to make the value negative. In either case, the hyphen appears at the front of the field after you terminate the field.

6. The alphanumeric representation of the field contents returns to the application program in the variables E$ and E4$(). If the value of the field is a valid numeric string (regardless of the field type), the numeric representation of the field's value returns to the application program in the variable Q. Otherwise, the value of Q is zero. The field number returns in the variable F. F is always the number of the field just processed. It may differ from the value of F that passed to the subroutine if you touched one of the SKIP-AHEAD or SKIP-BACK keys.

   If the field was a SPECIAL FUNCTION KEYS & EXEC ONLY field type, Q contains the number of the function that you used (0 through 31), 32 if you used EXECUTE, or 33 if you used EDIT. The value of E$ and E4$() may be indeterminate in this case.

In general, any field operation affects C, L, P, Q, R, T, E$, and E4$(), at least.

<u>DEFFN'35(F0$)</u> -- Display F0$ as error message on Line 24, sound audio alarm.

When the subroutine is called, the prompt "Error Message -" appears on Line 24, followed by the contents of the variable F0$ (the string passes to the subroutine), and the audio alarm (if any) sounds. The error message remains on Line 24 of the screen only until the operator touches a key or until another error message or operator message appears, whichever occurs first.

If the system error messages are off (if Error Message Display Flag F6$(14) equals N), the message does not appear.

<u>DEFFN'36</u> -- Display all fields starting at current field number F.

This routine displays the current contents of all displayable fields associated with the current screen, starting at field number F and ending with field number F0. F0 is the highest field number in the screen and is set during the loading of the screen's field parameters (using DEFFN'32, DEFFN'33). The return values are those described in DEFFN'37(Q), where Q=F0.

F is normally the number of the current field to be processed. Executing the subroutine does not change the value of F.

If F is zero, the first field to appear is field number 1.

DEFFN'37(Q) -- Display the contents of field number Q.

Input: Q -- Field number

Output: E$,E4$() -- Contents of field
Q -- Numeric value of field
R -- Row
C -- Column
P -- Position
L -- Length
T -- Type

This subroutine displays the contents of the field number specified in the subroutine call unless the field is one that does not permit display. The contents of the field return in the variables E$ and E4$(). If the field contains a valid numeric value, it returns in the variable Q, otherwise Q is zero.

DEFFN'38(O,E8$) -- (Internal) System error messages.

Input: O -- Message number
E8$ -- Information to include in error message

Output: F0$ -- Error message number O

This subroutine loads an array of error messages from file IDS2f000 into array E4$(). It selects message number O and puts it into F0$, puts the information in E8$ into F0$, then continues to DEFFN'35 to display message F0$.

DEFFN'39(F5$,P) -- Get limits of file F5$ on device number P.

Input: F5$ -- File name
P -- Device number used

Output: P -- Device number used
A -- Starting sector of file
M -- Ending sector of file
N -- Number of sectors used in the file
Q -- File status

Pass a file name (either a program file or a data file) and a disk device number to this subroutine. The system then gets the limits of the specified file from the specified disk, unless the file name begins with IDEAS2, IDS2, or ids2. In this case, the value of P changes to 2.

DEFFN'40(E8$) -- Load module E8$.

If the module being loaded is a cancel module (i.e., E8$= R3$(1)), the subroutine first checks the key buffer F1$() for any keys that may exist there. The existence of a key in the key buffer indicates to the system that a record was read and protected on disk. If there are any keys in the key buffer, the system rereads the indicated records, and rewrites each of them with the protect byte turned off.

It then loads the program specified in E8$ from device #4 starting at line 1000. (If the module name begins with IDS2 or ids2, the system loads the program from device #2.)

DEFFN'41(F5$,E$,J) -- Read a record from file F5$.

Input: F5$ -- Name of the primary or alternate key file
E$ -- The key associated with the desired record
J -- Record protect flag position within work buffer

Output: F$() -- Record contents
Q -- Return code - positive if record is found

This subroutine searches the specified file for the given key, reads the record, unpacks it, and places the record in the record buffer F$(), starting at Byte 1.

Whenever a data file is read, the value of J determines the placement of the record and the record protection as follows:

1. If J is less than or equal to -1, the record is not protected, and is read into the record buffer F$() and into the work buffer E$(), starting at the byte the absolute value of J represents.

2. If J is less than or equal to zero and greater than -1, the record is not protected and is read into the record buffer F$() only.

3. If J is greater than zero and less than 1, the record is protected and is read into the record buffer F$() only.

4. If J is equal to or greater than 1, the record is protected and is read into the record buffer F$() and into the work buffer E$(), starting at byte J.

The following chart summarizes the above conditions.

```
                                J
 -2           -1                0           +1          +2
              +   Don't put into E$()       +
 + Don't protect the record     +
                                +   Protect the record      +
```

If the record is unprotected when DEFFN'42 is loaded, the above is true and the value of Q that returns to the application program is positive.

If the record is protected when DEFFN'42 is loaded, the value of Q represents the station number of the station that protected the record. The record is still read and placed in the appropriate buffer(s), but cannot be updated. The keys are not written into the key buffer, nor does the current station protect the record. An error message appears that states that the record is currently in use. The message specifies the protecting station. The value of Q is negative.

If the system does not find the record, the value of Q is zero and an error message appears that states that the record is not in the file.

The above description of values for J and Q also applies to DEFFN'61 and DEFFN'62.

In general, file access may affect almost all of the numeric variables except F, F0, and R0. File access does affect E$, E5$, E6$, E7$, F3$, F5$, F9$, E0$(), E3$(), E4$(), and F$(), and may affect E8$, E9$, E$(), E6$(), F1$(), and F0$.

<u>DEFFN'42(F5$,J)</u> -- Write a record into file F5$.

Input:	F5$ -- Name of the primary or alternate key file J -- Record protect flag and position within work buffer
Output:	E3$() -- Record block containing record Q -- Return code - positive if record stored

Pass this subroutine a file name (F5$) and a position in the work buffer (J). The system copies the record from the work buffer, starting at byte J, into the record buffer (unless J=0, in which case the record is assumed to be in the record buffer F$() already).

The system then checks the primary and alternate keys (if any) for that record against the primary and alternate keys that may exist in the key buffer for the primary and any alternate files associated with the given file number. If a key exists in the key buffer, the PUT operation is an update, because a GET operation must have filled those key buffer positions.

If an update is sensed, each key in the record is compared to the corresponding key in the key buffer. If they are unequal, the old one is deleted from the file and the new one is inserted.

First a check is done for illegal duplicate keys. If an operation is a save, not an update, each key in the record is inserted into the appropriate key file.

The key buffer elements associated with the given file are then cleared, the record is packed, and it is saved in the primary file with the protect byte turned off.

If the record was stored successfully, a positive value returns to the application program in the variable Q. If the record was not stored successfully, an error appears that gives the reason (i.e., File full, or Illegal duplicate key) and specifies the file name in which the problem occurred. Any keys that were updated prior to the occurrence of the error condition are restored to their original condition.

<u>DEFFN'43(Q)</u> -- Retrieve contents of field by number.

Input: Q -- Field number

Output: E$,E4$() -- Contents of field
Q -- Numeric value of field or 0

Passing a valid field number to this subroutine causes the contents of the specified field to return to the application program in the variables E$ and E4$(). If the field represents a valid numeric value, the number value of the field returns in the variable Q. Otherwise, Q is zero.

<u>DEFFN'44(E$)</u> -- Retrieve contents of field by name.

Input: E$ -- Field name

Output: E$,E4$() -- Contents of field
Q -- Numeric value of field

Passing a valid field number to this subroutine causes the contents of the specified field to return to the application program in the variables E$ and E4$(). If the field represents a valid numeric value, the number value of the field returns in the variable Q. Otherwise, Q is zero.

<u>DEFFN'45(Q,E4$())</u> -- Put contents of E4$() in field number Q.

Input: Q -- Field number
E4$() -- Information to be displayed

Output: E$ -- Information displayed
Q -- Numeric value of field

Pass this subroutine a field number and an alpha string. The number of characters in the string equal to the length of the specified field will be copied into the field.

Unlike IDEAS Release 1, you cannot inhibit display of the field by passing a negative field number to DEFFN'45. This causes the system to crash.

The numeric representation of the string returns to the application in the variable Q if the string is a valid numeric string. Otherwise, Q is zero.

DEFFN'46(E$,E4$()) -- Put the contents of E4$() into field E$.

This subroutine gets the field number from the name and passes control to DEFFN'45 to process the field.

DEFFN'47 -- Get file attributes.

Input: V -- File number

Output: D -- Disk device number
H,L,Y -- Packed key length
P -- Pointer length
E -- Total KIE length
B -- Number of buckets
S -- Total sectors per bucket
T -- File type

Pass a file number to this subroutine. The file attributes return in the variables shown above.

DEFFN'48(C,E4$(),L) -- Prepare print buffer: Copy L bytes of E4$() to position C in output buffer E0$().

Input: C -- Position in print buffer
E4$() -- Text to be printed
L -- Length of text copied from E4$()

Output: E0$() -- Print buffer

If C is positive, the contents of L bytes of E4$() are placed in E0$, starting at byte C+1.

If C is zero, the contents of L bytes of E4$() are placed in E0$(), starting at the first blank that follows the last nonblank character.

If C is negative, the contents of L bytes of E4$() are placed in E0$(), starting ABS(C) bytes beyond the first blank that follows the last nonblank character.

If L is zero, L is considered the LEN of E4$().

The print buffer is E0$(), which consists of an array of 249 1-byte elements. When you initialize the print buffer, the first byte is set to HEX(01) and the rest are set to spaces.

DEFFN'49(L) -- Print a line of the print buffer.

Using the printer at device address #1, print the contents of the print buffer starting at byte 2 for L bytes, unless L is equal to zero. In this case, use the line length in Byte 1 of the print buffer as the length to be printed.

You can use this subroutine to generate a line feed on the printer.

DEFFN'50(H,W,E9$) -- Read an index sector.

Input: H -- Starting sector
W -- Redimension value (to 3x number of sectors to read)
E9$ -- Unchanged
D -- Device number

Output: E3$() -- Contents of sector(s) read

This subroutine redimensions array E3$() into the size passed to it in W (W/3 sectors) and redimensions E0$() into a 1-sector size. E3$() is then filled from disk, beginning at sector H. E9$ is reset to the value passed in the DEFFN'50 call.

DEFFN'51(E4$()) -- Complement and pack key.

Input: E4$() -- Unpacked and uncomplemented key

Output: E4$() -- Packed and complemented (if applicable) key

This subroutine is called within DEFFN'41 (and DEFFN'79) to read a record and within DEFFN'60 to set up a sort array and find the first logical record. The key passed to it in E4$() is complemented and packed and returned in E4$(). E$ is not affected.

DEFFN'52 -- Key access (part of DEFFN'59 that sequential processing uses).

Input: E$ -- Key from '59

Output: E3$() -- Index block
F9$ -- Key position

WARNING

This subroutine is internal to other IDEAS subroutine modules. The developer should not use this subroutine.

This subroutine exists as a separate subroutine from DEFFN'59 because there are situations where the sequential processing subroutines are loaded into a local partition and the remainder of the subroutines are loaded into a global partition. Many variables must be set to specific values before you call DEFFN'51 or indeterminate and possibly disastrous results may occur.

For example: You must have previously used DEFFN'47.
N must be a valid bucket number.
X must be 6.

Using DEFFN'50, the subroutine loads 8 sectors of the key portion of the data file into E3$() and then searches it for the key named in E$, the position of which is given in F9$.

<u>DEFFN'53(F0$)</u> -- Display message with Attention on Line 24.

Input: F0$ Message

The word "Attention":, followed by the string passed to the subroutine (up to 64 characters), appears on Line 24 and remains there until the operator touches the next key or another message overwrites it.

<u>DEFFN'54(E$)</u> -- Get field number given field name.

Input: E$ -- Field name

Output: Q -- Field number

The subroutine loads the list of field names for a screen or report into array E3$(), if necessary, and searches for the name in E$. The field number returns in Q. F5$ and E9$ are affected in this operation. DEFFN'54 also gets field numbers for the @TSTFLDs, file names, screen name, and @SYSBUF0.

<u>DEFFN'55(N,A)</u> -- Round and put a numeric field into work buffer.

Input: N -- Field number
A -- Value to be rounded

Output: E$,E4$() -- Alpha value of field
Q -- Numeric value of field
H -- =0 for fail condition if overflow occurs

This subroutine rounds the value passed to it as A to the number of decimal places you specify in the field parameter record for field number N. It then converts the value to an alpha string with justification and the fill character that the field parameters specify. It then places the string in field number N in the work buffer. If the field parameters allow the field to be displayed, this subroutine displays it.

This subroutine provides a fail condition (H=0) in case of overflow (i.e., if the value is too large to fit into the field). The value of H is not set unless an overflow is encountered (generated programs set it to pass). Therefore, to use DEFFN'55 you must set H = 1 before calling DEFFN'55 for an overflow test.

The alpha representation of the field returns to the application program in the variables E$ and E4$(). The numeric value returns in the variable Q. In the case of an overflow, the return value Q is also set to zero and both E$ and E4$() contain an image string (-###.##, for example).

If you specify a field as left-justified and you allow plus or minus signs, a positive number begins at Byte 2 in the field because the first character is reserved for the sign.

Signed field types are 1 and 3 and any alphanumeric field. If you try to place a negative number into an unsigned numeric field (Types 0 and 2), the absolute value of the number is placed in the field.

You can use only real fields as destinations for DEFFN'55. You cannot use file names, screen names, @TSTFLDs, and @SYSBUF0.

If you use a single character alphanumeric field for DEFFN'55, a crash results. IDEAS generates a format and uses a PRINTUSING TO statement to place the value in the field. An alphanumeric field is considered signed and the first position is reserved for the sign, so a single character field causes a crash because of a missing # in the format variable.

<u>DEFFN'56(E8$,A)</u> -- Date validation

Input:	E8$ -- Alpha representation of date A -- Format of date in E8$
Output:	E$ -- Numeric representation of date as shown below Q -- Julian date, or zero if date was invalid H -- H = 1 : Pass, H = 0 : Fail

This subroutine allows the operator to enter a date as an alphanumeric string in any of six formats. If the date passed is valid according to the specified formats, a pass condition results (H=1). If not, an error message appears and a fail condition results.

Format codes:
1. MMDDYY (example: 021782)
2. DDMMYY (example: 170282)
3. MMMDDYY (example: FEB 02 82)
4. DDMMMYY (example: 17 FEB 82)
5. YYDDD (example: 82048)
6. DDDDD (example: 29998)

DEFFN'57(E8$,A,B) -- Convert alphanumeric date in format A to format B.

Input: E8$ -- Alpha representation of date
A -- Format of date presented
B -- Format of date desired

Output: E$ -- Numeric representation of date as shown in '56
Q -- Julian date, or zero if date was invalid

Pass a date in one of the allowed formats (see DEFFN'56) to E8$. Pass the format in which the date currently exists to A. Pass the format in which you want the date to B. The subroutine converts the date. The date appears in E$ in the new format. The subroutine also performs validation.

DEFFN'59(V,J,E$,X) -- Key file access subroutine (insert/retrieve/delete).

Input: V -- File number of key file to access
J -- Protect flag:=0, do not protect record; > 0, protect
E$ -- The key to be inserted, retrieved, or deleted
X -- Operation Code

Output: Q -- Returns zero if operation unsuccessful; positive if successful; negative if using station is found already in use
D -- Disk device number where record located
H -- Starting sector number
Y -- Length of key
E0$() -- Coarse key pointer
E3$() -- Fine key pointer
F3$ -- Pointer

This subroutine (or portions thereof) is used throughout all of the access routines and performs a number of functions.

X is an operation code that implies the type of action required:

0 -- Find a key
1 -- Insert a new key without checking for duplicates
2 -- Insert a new key if no duplicate exists
3 -- Delete a key

Because of problems that might occur through the indiscriminate use of this subroutine, you should avoid it. The higher level file access subroutines handle all of the appropriate file maintenance.

If the specified operation is not successful, a value of zero returns to the application program in the variable Q. If the record you want to use is currently in use at another station, the value of Q is the negative of the station number of the using station.

If the operation is successful, the value of Q is positive, the disk device number of the platter containing the record is D, and the sector number where the record starts is H.

<u>DEFFN'60(F5$,E$)</u> -- Set up sort array.

Input:	F5$ -- File name E$ -- Starting key
Output:	E6$() -- Sort array C -- Length of key

This subroutine reads the first key and pointer greater than E$ from each bucket in the file and places it in ascending sort order in array E6$(). It sets up the merge array so that you can begin processing with the lowest key that is equal to or greater than the value passed to the subroutine. It does not actually retrieve any records. It merely sets up the sort array so that the application can process the file by using FIND NEXT (DEFFN'62) for each record, rather than having to perform a FIND FIRST (DEFFN'61) followed by a series of FIND NEXT operations.

Each element in E6$() consists of a string variable of C+6 bytes where C is the length of the actual key. The three bytes immediately following the actual key represent the pointer associated with that key, and the last three bytes are the key's sector number and the key's byte number within that sector.

This subroutine also affects E3$() and E0$().

<u>DEFFN'61(F5$,E$,J)</u> -- Find first logical record equal to E$.

Input:	F5$ -- File name E$ -- Starting key J -- Protect flag
Output:	Q -- Record return code F$() -- Record

Sets up the sort array as in DEFFN'60, and retrieves the record that corresponds to the lowest key in the merge array.

The protect flag J and the return code Q operate as described above for DEFFN'41.

DEFFN'62(F5$,M,J) -- Find next logical record in file F5$.

Input: F5$ -- File name
M -- Previous key deletion flag (see below)
J -- Protect flag/position indicator (see below)

Output: Q -- Record return code
F$() -- Record

Use this file only if you used DEFFN'60 or DEFFN'61 for the same file previously. Otherwise, a fatal error can occur. You cannot use sequential processing operations on more than one file at a time.

For most normal sequential processing operations, the value of M should always be 1. If you are processing the file sequentially to delete a certain record, however, the value of M must be zero the next time you use DEFFN'62 after you delete the record immediately preceding the same files.

For a description of the operation of the protect flag J and return code Q, see above at DEFFN'41.

DEFFN'63(E8$,F0$,E8$(1)) -- Display 80-byte message at (24,0).

Input: E8$ -- First 16 characters of message
F0$ -- Remainder of message
E8$(1) -- Flag to sound audio alarm

This subroutine allows you to create a message or error message of up to 80 characters for display on Line 24 of the screen that remains on the screen until the user touches a key. The first 16 characters of the message go into the variable E8$ and the remainder go into F0$. If the character ! is passed to E8$(1), the audio alarm (if any) sounds. If any other character is used, it does not sound.

DEFFN'64(T,E) -- Copy a field by number.

Input: T -- Number of field to be copied
E -- Number of field to receive information

Output: STR(E$(),A,B,) -- Information copied

When this subroutine is called, it uses DEFFN'43 to get the contents of field T, which it then copies to field E.

This subroutine affects the values of P, L, R, and C.

DEFFN'65(F7$,E7$) -- Copy a field by name.

Input: F7$ -- Field to be copied
E7$ -- Field to receive information

Output: STR(E$(),A,B,) -- Information copied

This subroutine uses DEFFN'54 to get the field number of fields F7$ and E7$. It then passes to DEFFN'64 and performs in the same way as DEFFN'64.

You can also use DEFFN'64 and DEFFN'65 to copy @TSTFLDs, files, and @SYSBUF0.

This subroutine affects the values of P, L, R, and C.

DEFFN'66(V,C) -- Unpack record in E3$() to E3$() for File V.

Input: V -- File number
E3$() -- Packed record
C -- Starting position in buffer

Output: F$() -- Unpacked record

If the record is contained within the file buffer E3$(), pass the starting position of the data record to C. This causes DEFFN'66 to copy the record from E3$() to F$() and unpack it.

If C is zero, the packed record must be in F$() already, starting at Byte N (where N is the integer of the number of packable numeric characters in the record format specification divided by 2).

DEFFN'67(V,F3$,X) -- (Internal) Get record, given pointer.

Input: V -- File number
F3$ -- Key pointer
X -- Operating code

Output: E3$() -- Record block

This is an internal subroutine that reads a record given the file number V, the key pointer F3$, and an operating code X. If X <> 0, the subroutine returns immediately.

DEFFN'69(V) -- Form key, given record and file number.

Input: V -- File number
F$() -- Record buffer

Output: E4$() -- Complemented key
E$ -- Uncomplemented key

The record must already be in the record buffer, F$().

This subroutine builds the key for the specified file (ABS(V)). The key can consist of up to 5 fields. It complements any field that you specify to be in descending sort order.

The correct, complemented key is left in the variable E4$(). The uncomplemented key is left in the variable E$.

DEFFN'78(F5$) -- Set V equal to the file number of the file F5$ names.

Input: F5$ -- File name

Output: V,Q -- File number

Passing the name of a currently open primary or alternate file to this subroutine sets the variables V and Q equal to the file number for that file. If the file you specify does not exist or has not been opened, the value of V is zero and an error message appears.

DEFFN'79 -- Get a record (same as DEFFN'41, but does not get the file number).

This is the core of DEFFN'41, after DEFFN'78 has retrieved the file number. The operation of '79 can be found in '41.

APPENDIX F
SPECIALIZED IDEAS MODULES

F.1 IDEAS RUNTIME FILES

The following files are the IDEAS Release 2 files that you need to run an IDEAS-developed application.

IDS2PIX0	IDS2PUM1	IDS2PBX6	IDS2SUB2	IDS2sR11
IDS2PIX1	IDS2PRX0	IDS2PBX7	IDS2SUB3	IDS2sR12
IDS2PIX2	IDS2PRX1	IRS2PR20	IDS2SUB4	IDS2sR13
IDS2PIX3	IDS2PRX2	IDS2PR21	IDS2SUB5	IDS2sR16
IDS2PIX4	IDS2PRX3	IDS2PR25	IDS2SUB6	IDS2sR17
IDS2PIX5	IDS2PRX4	IDS2PR26	IDS2SUB7	IDS2sR18
IDS2PIX6	IDS2PBX0	IDS2PR27	IDS2SUB8	IDS2f000
IDS2PIX7	IDS2PBX1	IDS2GLOB	IDS2MX01	IDS2f001
IDS2PIX8	IDS2PBX2	IDS2GLBL	IDS2PPX0	ids2f001
IDS2PIX9	IDS2PBX3	IDS2LOCL	IDS2s006	IDS2f005
IDS2P003	IDS2PBX4	IDS2SUB0	IDS2sI04	IDS2PER1
IDS2P006	IDS2PBX5	IDS2SUB1	IDS2sP01	

F.2 IDEAS UTILITY MODULES

For each of the supplementary utilities and development utilities, you need certain IDEAS modules to run the utility from an application menu. All of these modules are Type I programs, except IDS2PU76, which is Type X. Table F-1 lists the names of each utility program and the additional IDEAS modules needed to run the utility.

Table F-1. IDEAS Utility Modules

Utility	Utility Program	Necessary Additional Modules
Protect All Records	IDS2PU01	IDS2sSC0
Release All Records	IDS2PU99	IDS2sSC0
Change Device Addresses From a Menu Stand-alone	 IDS2PU03 IDS2PU76	 None IDS2PU77 IDS2P003 IDS2PU03 IDS2sU03 IDS2P001 IDS2P004 IDS2P012 IDS2LOCL IDS2SUB1 IDS2SUB2 IDS2SUB3 IDS2SUB4 IDS2SUB5 IDS2SUB6 IDS2SUB7 IDs2SUB8 IDS2GLOB IDS2GLBL IDS2PUM1
Convert IDEAS File To Wang TC File	IDS2PU13	IDS2PU05 IDS2PU06 IDS2PUIa IDS2PU11 IDS2sU05 IDS2SUB1 IDS2SUB2 IDS2SUB3 IDS2SUB4 IDS2SUB5 IDS2SUB6 IDS2LOCL IDS2P001 IDS2SUB8
Convert Wang TC File To IDEAS File	IDS2PU14	IDS2PU07 IDS2PU08 IDS2PU11 IDS2sU07 IDS2SUB1 IDS2SUB2 IDS2SUB3 IDS2SUB4 IDS2SUB5 IDS2SUB6 IDS2LOCL IDS2P001
Expand Size of IDEAS File, Saving Data	IDS2PU12	IDS2PU05 IDS2PU06 IDS2PU07 IDS2PU08 IDS2PU09 IDS2PU10 IDS2PU11 IDS2PU13 IDS2PU14 IDS2sU05 IDS2sU07 IDS2sF11 IDS2PF11 IDS2PF12 IDS2PF14 IDS2P001 IDS2P004 IDS2SUB1 IDS2SUB2 IDS2SUB3 IDS2SUB4 IDS2SUB5 IDS2SUB6 IDS2SUB8 IDS2LOCL
Key File Recovery	IDS2PU88	IDS2PU89 IDS2PU90 IDS2SUB1 IDS2SUB2 IDS2SUB3 IDS2SUB4 IDS2SUB5 IDS2SUB6 IDS2LOCL IDS2P001 IDS2PF14 IDS2sU88 IDS2sU90 IDS2sU89

Table F-1. IDEAS Utility Modules (continued)

Utility	Utility Program	Necessary Additional Modules
Key File Record Dump	IDS2PU21	IDS2PU17 IDS2PU18 IDS2PU19 IDS2PU20 IDS2PU24 IDS2P001 IDS2P002 IDS2P004 IDS2sSCO IDS2sU13 IDS2sU14 IDS2SUB1 IDS2SUB2 IDS2SUB3 IDS2SUB4 IDS2SUB5 IDS2SUB6 IDS2SUB7 IDS2SUB8 IDS2SUB9 IDS2LOCL
CPU System File Maintenance	IDS2PUM2	IDS2sUM1
Device Addresses Entry From Primary Program Selection Menu	IDS2MU02	

F.3 IDEAS FILES THAT YOU SHOULD NOT COMPRESS

Nine IDEAS modules exist that you should not compress using the Wang Integrated Support System (ISS) software package. These modules are:

IDS2SUB1	IDS2SUB3	IDS2SUB5	IDS2SUB7	IDS2SUB9
IDS2SUB2	IDS2SUB4	IDS2SUB6	IDS2SUB8	

F.4 IDEAS FILES REQUIRED TO RUN DATA FILE INITIALIZATION

The data file initialization program module is IDS2PF21. If you want to run the data file initialization utility from an application menu, you must include six additional IDEAS files with the application system, in addition to IDS2PF21. These files are:

IDS2PF13 IDS2PF21 IDS2PF00 IDS2sF13 IDS2PF14 IDS2sSc0

APPENDIX G
REMOTE TRANSMISSION

G.1 INTRODUCTION

IDEAS includes special pass/fail (P/F) options that you can use to establish a telecommunications (TC) interface. When used in conjunction with Wang 2200 3271 Emulation software, an IDEAS-developed program can transmit and receive unformatted records in 3270 protocol across a communications line. To use this feature, you must be familiar with the operation of the Wang 2200 3271 Emulation software.

This appendix includes three major sections. The first section provides a summary of the hardware and software requirements for interfacing an IDEAS application with an emulated 3271 communications cluster. The second section describes how to incorporate communications capabilities into an IDEAS-developed application. The final section details how to modify IDEAS-supplied code to accommodate particular communications needs.

G.2 ENVIRONMENTAL REQUIREMENTS

In order to run an IDEAS-developed program that uses communications options, you must configure an emulated 3271 cluster and activate the 3721 software before you run the IDEAS program. Complete instructions for these procedures are provided in The 2200 3271 User Guide and The 2200 3271 BSC Emulation Implementation Guide and User Manual. A summary of the requirements follows:

- The TC controller device selected in the IDEAS application START module must be available throughout the application.
- The line of connection to the host must be up and available.
- You must load and run the 3271 Emulation software to configure and activate a cluster, including the appropriate microcode in a 2228D TC board.
- You must load the universal program for 3271 emulation into Partition 1, a background partition assigned to any terminal. Partition 1 must be between 2K and 5K bytes.
- You must load and run the Master Controller Task for 3271 emulation in Partition 2, a background partition assigned to any terminal.

G.3 DEVELOPING A TC APPLICATION

To incorporate TC capability into an IDEAS-based application, you must use P/F option K, L, M, or N on the Pass/Fail Action Specification screen (Figure 5-12). Using one of these options overlays a Wang-supplied or user-modified TC module onto the generated code. This module communicates with the universal program in Partition 1 and with the Data Link Processor (DLP) microcode loaded into the communications controller. The universal program in turn communicates with the MCT in Partition 2, which then transmits to the waiting host.

There are two subdivisions in this section. Section G.3.1 details the implementation of the TC options of the Pass/Fail Action Specification screen. Section G.3.2 describes some aspects of TC that you should take into account when you define your application.

G.3.1 Implementation

The specification for a TC transmission is a P/F option chosen from the Pass/Fail Action Specification screen (Figure 5-12). As with all other P/F options, four fields appear on the screen when you define a communications action. Each of the four TC options uses these fields in the same way.

In the first P/F field, as with all other P/F options, you must specify under what condition you wish to perform the transaction by indicating P (pass), F (fail), or B (both).

In the second field, you must indicate which action specification you wish to perform. There are four communications options:

K -- Transmit from Operand 1

L -- Transmit indirect

M -- Receive to Operand 1

N -- Receive indirect

Refer to Section 5.4 for an explanation of indirect actions.

Operand 1 must contain the name of the field, @TSTFLD#, file name, FILE # n, or screen name that is to be transmitted or into which the received message is to be copied. Specify a screen name to transmit or receive the entire contents of the work buffer. For indirect operations, this operand must contain the name of the field that contains the name of the field, @TSTFLD#, file name, FILE # n, or screen name to be transmitted.

Operand 2 must contain the name of the IDEAS TC interface module, IDS2327U, or the name of a developer-modified TC interface module. This module is appended to the generated application code at execution. For information on modifying the TC interface module, refer to Section G.4.

The only TC transactions permitted are the four listed above. There are circumstances in which a transmission always expects a response. You must specify this by using two separate TC transactions: one for transmission followed by one that receives the response.

If an error occurs in the communications link, IDEAS sets a fail condition and returns a system error code to the calling application, resetting both the previous P/F condition and the system error code. If a communications link is successful, IDEAS sets the P/F state to pass but does not change the previous system error code. You can test both the P/F condition and the system error code in the application logic. The communications interface does not change the value of the application-accessible SYSFLAGs.

If a TC link fails, you can branch back to the same P/F action and try again. In this case, you need to recall the previous P/F condition or the previous error message. To account for this situation, you must use the following extra conditions and operations in the P/F actions immediately preceding and following the TC transaction; refer to Section 5.6 for instructions on implementing these conditions and operations.

SETOLDPF -- Saves an existing P/F condition prior to executing a P/F that will change the current state

GETOLDPF -- Recalls the P/F condition that was saved by means of SETOLDPF

IFOLDPFP -- Tests the P/F condition saved by means of SETOLDPF
IFOLDPFF
OROLDPFP
OROLDPFF

IFERR### -- Tests for the last system error message number
ORERR###

In addition to the above operations, you can also use the variable ERRORMSG or @ERRMSG# as Operand 1 in P/F Option I or J to copy the last system error message or error message number into any valid destination field.

G.3.2 Considerations

There are many specific issues you should consider when you plan TC interface. Some of these concerns are:

- Most communications receptions require some time before the host responds. Consider whether you wish to enable the CANCEL key during communication.

- If there is not sufficient memory in the partition running the application to load the TC interface module, a program crash occurs. Test this condition before you install an application.

- Perform a receive action (Option M or N) before you perform a transmit action (Option K or L). Typically, the host computer sends some information as soon as the line is established and it is necessary to process this before any transmission takes place, even if such processing consists of only copying the data to BLANKFLD (throwing it away). If there is no record in the receive buffer, an error message normally appears, but you can suppress this error message so that it is transparent to the operator.

- The IDEAS/TC interface does not distinguish among expected responses, unsolicited messages (responses that may be left in the receive buffer if an operator cancelled out of a prior transaction), or broadcast messages. If a transmission is attempted while there is a message in the receive buffer, the transmission fails and an error code indicates that the pending message must be processed first. You must define your application to account for this possibility by reading the message and processing it before attempting the transmission again.

- An IDEAS program supports only one protocol and only one TC board in any given application program. An application can support more than one protocol or the same protocol on more than one TC board, but only if the different protocols and/or TC boards are addressed from separate IDEAS program modules.

- As indicated in Section G.3.1, you may specify only a single field, file, or screen in Operand 1 for transmission or reception. If you wish to append a standard header string or other information to the front of each transmission record, you must do this in either of two ways:

    -- Assemble the complete transmission record including the header in one of the available buffer fields, files, or screens by means of the Set a Field operation (described in Section 5.3) before invoking the TC transaction.

    -- Modify the TC interface program module to append the header before each transmission. Refer to Section G.4 for instructions about modifying TC code.

If you wish to test or process certain bytes or strings in a received message, you must set up these bytes as individual fields within the received record entry.

G.4 CODE MODIFICATION

As described in Section G.3, when you specify a TC transaction on the Pass/Fail Action Specification screen, you must indicate the name of the IDEAS TC interface module. The IDEAS-supplied module, IDS2327U, enables you to transmit and receive unformatted records in 3270 protocol. A lowercase version of this module -- ids2327u -- is also supplied with the IDEAS code. This module is a noncompressed version of IDS2327U with extensive internal documentation. You can use this program as a basis for a TC interface module that meets the needs of a particular application.

Like the IDEAS-supplied module, your modified version of the TC interface module should begin at Line 3500 and provide system error messages and appropriate return codes to allow you to specify appropriate P/F actions based on the success or failure of a given transaction. Assign your own program name to the completed module, leaving the IDEAS-supplied module intact on the system utilities disk. You can then specify your program name as the IDEAS TC interface module on the Pass/Fail Action Specification screen.

Some modifications you can make to the TC interface module are:

- Changes to the messages that appear during the communications cycle to inform the operator of what is taking place.
- A standard identification message appended to the beginning of any message transmitted to a host.
- A standard retry or time-out specification.
- Provision for nonstandard operator intervention during the communications cycle.
- Maintenance and/or testing of a specified sequence number indicator for properly identifying transmission/response record pairs.
- Selection of a TC board address different from the one the IDEAS application START module selects, in case an application supports different TC boards or protocols from different program modules.
- Interception and/or processing of logic or application errors that occur in a response message that the host detects and flags but the general interface structure does not. This may occur if the response messages are not valid to the host but are valid to the standard interface module.
- Setting IDEAS SYSFLAGs to return more information to the calling application than the standard interface module normally provides.

- Providing an automatic "transmit and wait for response" operation where such a situation may be the only applicable case in a specific application. In this case, only the TRANSMIT action specification (P/F Option K or L) is valid and you must provide the code to handle the proper disposition of the received data.

APPENDIX H
IDEAS PROMPTS AND ERROR MESSAGES

Message Number	Message
0	This is a required field. Please enter a non-blank value.
1	Invalid character - must be a digit.
2	Invalid character - must be a digit or a sign (+ or -).
3	Invalid character - must be a digit or a decimal point.
4	Invalid character - must be a digit, a sign, or a decimal point.
5	Invalid character - must be an upper case letter.
6	Invalid character - must be an upper case letter or a digit.
7	Invalid character - must be upper case, numeric, or punctuation.
8	Invalid character - must be any alphanumeric or punctuation.
9	Invalid character - must be "Y", "y", "1", "N", "n", or "0".
10	Invalid character - must be a function key, EDIT, or EXECUTE.
11	Module "########" selected for loading not currently available.
12	Time default & format test functions not currently available.
13	Invalid date - date may not be #.
14	File ######## not currently open.
15	Specified record not found in file #.
16	Illegal duplicate key in file #.
17	File ######## is full.
18	Record is currently protected by station #.
19	End of file ########.
20	You are not authorized to write to file ########.
21	Blank record may not be saved.
22	You are not authorized to read from file ########.
23	If this field is used, all character positions must be non-blank.
24	Invalid numeric field.
25	Insufficient memory available to execute this application.
26	You are not authorized to execute this application.
27	Invalid password. Re-enter or touch FN '31 to return to menu.
28	Must be 'P' (Pass), 'F' (Fail), 'B' (Both) or blank.
29	Action specification must be blank or none of those listed.
30	Invalid specification for reports!
31	Invalid specification.
32	Field already exists -- please re-enter.
33	Field name is reserved for system use -- please re-enter.
34	Field ######## does not exist - please re-enter.
35	Position of sub-field must be between 1 and #.
36	Field length must be less than 256.

Message Number	Message
37	Length must be between 1 and #.
38	Field type must be between 0 and #.
39	Number of decimals must be less than field length.
40	Number of decimal places must be 0 to 7.
41	Security user class must be 0 - 9 or A - F.
42	Security user class must be greater than or equal to #.
43	File number must be between 1 and #.
44	Disk error on device # -- check disk or FN '31 to cancel.
45	Invalid specification - Field 1 must be before Field 2.
46	Address is missing from IDEAS device list. Re-enter or CANCEL.
47	Number of volumes must be 1 - 8.
48	Number or records cannot be zero.
49	Use '1 to change addr. of first volume of multi-volume file.
50	Total sectors allocated must equal total sectors required.
51	You may not allocate more sectors than are available.
52	Key & record combination is too large for this file type.
53	Space in this field is reserved for a sign and/or decimal point.
54	Security file is not open. Press FN '31 to cancel.
55	Too many records for # volume(s) -- please re-enter.
56	Invalid pointer in data file. Please notify WANG support center.
57	Files beginning with "IDS2" must be single volume.
58	Touch EXECUTE to accept, or follow instructions above.
59	Disk allocation failed.
60	Must be 0 or one of the file numbers shown above.
61	######## must be 0, and column plust length must be 82.
62	Length must be from 1 to 255.
63	Please enter only '1' or '2'.
64	Please enter only '1', '2', or '3'.
65	Press EDIT to modify or EXEC to accept.
66	######## is a scratched data file. Please re-enter.
67	######## is a scratched program file. Please re-enter.
68	######## does not exist. Please re-enter.
69	######## is an active program file. Please re-enter.
70	CPU ID number doesn't match partition. Call Wang support center.
71	Device ### is not available. Please check status or address.
72	### is not in the 2200 master device table. Please re-enter.
73	Disk error ###. Make sure the device is ready.
74	Unanticipated error ##. No recovery has been programmed.
75	File ######## is damaged and cannot be read.
76	Essential address ### is missing from the IDEAS device table.
77	Volume # of file is missing. Please cancel.
78	Alternate control file ######## doesn't exist. Please cancel.
79	Alternate file ######## is missing. Please cancel.
80	E1$() is not dimensioned properly. Call Wang support center.
81	TC file is damaged. Please cancel.
82	Processing record number
83	Invalid system configuration (3270 universal global not present).
84	Telecommunications controller not active.

Message Number	Message
85	Invalid address (###) selected fro telecommunications controller.
86	Communications line is not available.
87	Communications transaction cancelled at operator's request.
88	Communications reception failure. Expected message not present.
89	Communications reception failure. Transmission expected.
90	Communications transmission failure. Received message waiting.
91	Communications failure. Expected response not properly received.
92	File is not an IDEAS 2 data file.
93	Type 1 files cannot be protected.
94	Alternate file -- do you want primary file?
95	End of file has been damaged. Correct and re-start.
96	File address is incorrect. Correct and re-start.
97	File is not an IDEAS 2 data file -- or wrong device address.
98	File is not at address indicated in control file.
99	The 'START' module named is not at the address given.
100	'START" module location incorrect. Correct and re-start.
101	Control file ######## location incorrect. Correct and re-start.
102	This is an illegal address selection. Please re-enter.
103	Band screen file ######## on disk -- touch 'EXEC'
104	Screen file address incorrect. Correct and re-start.
105	File must be opened first. Touch any key ton EXIT.
106	Control file is not located at correct address.
107	File is not located at correct address. Correct and re-start.
108	Volume not at indicated address. Correct and re-start.
109	Types 1 and 4 files cannot be recovered.
110	Alternate key file -- Enter primary file name.
111	Cannot reconstruct. Cancel and use TC conversion.
112	Illegal duplicate key from alternate file will erase record.
113	Printing and deleting record with illegal duplicate key.
114	Touch EXECUTE to return to menu, EDIT to enter new file name.
115	All required changes have been made.
116	Touch EXECUTE to accept and save all changes, EDIT to modify, FN '31 to CANCEL.
117	Touch EXECUTE to accept and save the changes, EDIT to modify, FN '31 to CANCEL.
118	Touch EXECUTE to accept and save ALL changes, EDIT to modify, FN '31 to CANCEL.
119	Processing file ########
120	######## is not at disk address for control files. Correct and restart.
121	Address of vol. ## duplicates another volume. Touch 'EXEC."
122	Correct file ######## volume addresses manually when finished. Touch 'EXEC."
123	######## not at address for control files. Correct and restart.
124	ID must be your ID or blank.
125	Touch EXECUTE to accept all inputs, EDIT to modify, FN'31 to CANCEL.

Message Number	Message
126	Control file ######## is not located at correct address -- touch 'EXEC.'
127	Data file ######## is not located at correct address -- touch 'EXEC.'
128	Key SF'0 to bypass this file or SF'31 to cancel batch update.
129	Touch EXECUTE to begin key file recovery, EDIT to modify, FN'31 to cancel.
130	Sector ##### has been damaged and is being reconstructed.
131	Touch EXECUTE to return to menu, '23 to continue with Key File Recovery.

PART II
IDEAS RELEASE 2 FAMILIARIZATION GUIDE

This exercise is meant to introduce you to the IDEAS Release 2 utilities. The exercise creates a simple data entry program that uses one data file and one data entry screen and defines one simple report. It does not use security options, multiple data files, subscreens, and other IDEAS Release 2 features. In this exercise, you will create the following:

1. Customer Data File - MAILLIST
2. Data Entry Screen - MAILSCRN (with Edit Specifications)
3. START Module - MAILSTRT
4. Main Menu Module Without Password Security - MAILMENU
5. Customer Maintenance Program - MAILPROG
6. Customer Report Module - MAILREPT

At the conclusion of this exercise, you should be able to run the START program MAILSTRT that opens the data file MAILLIST and displays the menu MAILMENU. MAILMENU should display a menu with two entries: one to run the data entry program MAILPROG that uses the screen MAILSCRN, and the second to print the report defined in the program MAILREPT.

1. Before you can run the IDEAS utilities, IDEAS must be configured as described in detail in Section 2.3 of The IDEAS Release 2 User Manual, either in local mode or in global mode. To run in local mode, your partition must be at least 28K. To run this exercise in global mode, load the global subroutine IDS2GLOB into a 14K partition in the same bank from which you access the IDEAS utilities. Running this program should release the terminal, allowing you to run IDEAS in a foreground partition. For this application, 14K should be sufficient memory in the foreground partition.

   Throughout the IDEAS utilities, the screen prompts indicate that the you should touch EXECUTE to accept a screen. This instruction applies to a Model 2236DW terminal. If you have a Model 2236DE terminal, touch the RUN key to accept a screen. On both types of terminals, the RETURN key also works to accept a screen.

   Keep in mind that all fields on the System utilities screens terminate automatically (the cursor moves to the next field) when full. Of course, if a field entry is less that the full size of the field, you must touch the RETURN key to make that entry.

   You can control cursor movement on all system utilities screens with FN 04, 07, and 11 through 14 (the cursor movement keys along the top of the terminal keyboard). You can use these keys to backspace to previous fields or to skip several fields at a time when editing a screen. On a 2236DW terminal, the cursor movement keys are equivalent to FN 04, 07, 12, and 13.

2. Enter LOAD RUN "IDEAS2" and the IDEAS Disk Address Selection screen, Screen 1, appears.

```
I.D.E.A.S.          Inquiry, Data Entry, and Access System          Release 2.1

            This module requires that you specify the disk address
            where the I.D.E.A.S system utilities reside.  The disk
            addresses supported by I.D.E.A.S. are listed below.

            Please choose the appropriate address for your system
            from those listed and touch RETURN, or just touch the
            RETURN key to accept the default address as shown.

            The disk address you choose (if valid) will become the
            new default address for this module on this terminal.

          ------< Valid disk address supported by I.D.E.A.S. >------
          310  320  330  B10  B20  B30  350  360  370  B50  B60  B70
          D10  D11  D12  D13  D14  D15  D50  D51  D52  D53  D54  D55
          D20  D21  D22  D23  D24  D25  D60  D61  D62  D63  D64  D65
          D30  D31  D32  D33  D34  D35  D70  D71  D72  D73  D74  D75

          Please enter I.D.E.A.S. system utilities disk address. D23
```

Screen 1. IDEAS Disk Address Selection Screen

3. The lower right corner of Screen 1 displays the disk address where the IDEAS utilities reside. If the address is incorrect, enter the correct one. Touch EXECUTE to continue and the IDEAS Security Access screen (Screen 2) appears.

```
IDEAS System Utility - Development Access Security Module          Release 2.1


                    User Identification Code     ■■■

                    User class     Password
```

Screen 2. Security Access Screen

4. The first time you run the IDEAS utilities after installation, no ID or password is required. You enter a security file management utility directly and you are asked to establish a user who will be a security administrator, with a user class of F. For information on security administration, refer to Subsection 2.3.3 of <u>The IDEAS Release 2 User Manual</u>.

   If you are not the first user to run the IDEAS utilities, a security administrator on your system must establish your user ID and password. If this has been done, enter your user ID and password at this point.

5. If you are running in global mode and a system date was entered already, the next screen that appears is the Primary Program Selection screen (Screen 5); continue with Step 7. Otherwise, the System Date screen (Screen 3) appears. Initially, the date field is blank. Follow the instructions shown on the screen to enter today's date in the format mmddyy. This date is carried throughout your work on the system and is used in the documentation and program creation modules.

```
IDEAS System Utility - System Date Edit Module                    Release 2.0



        Please enter today's date in the form mm/dd/yy ■■/  /
```

Screen 3. System Date Screen

If the date is correct, touch EXECUTE.

6. If device addresses for the peripheral devices are set already, the Primary Program Selection Menu screen (Screen 5) appears; proceed with Step 7. Otherwise, the Peripheral Device Selection screen, Screen 4, appears.

IDEAS System Utility - Peripheral Device Selection Module	Release 2.1

Device # 01 / 216	Printer
Device # 02 / D23	IDEAS system utilities
Device # 03 / D21	Screens & program control files
Device # 04 / D21	Application program files
Device # 05 / D21	Data file control files
Device # 06 /	Telecommunications controller
Device # 07 / D21	Application data files
Device # 08 / D21	Application data files
Device # 09 / D21	Application data files
Device # 10 / D21	Application data files
Device # 11 / D21	Application data files
Device # 12 / D21	Application data files
Device # 13 / D21	Application data files
Device # 14 / D21	Application data files
Device # 15 / D21	Application data files or other

Touch FN key corresponding to device to be changed, EXECUTE to accept as shown ■

Screen 4. Peripheral Device Selection Screen

Be certain that addresses for Device Numbers 1 through 7 are correct for your system configuration. If you need to change any of the addresses, touch the function key that corresponds to the device number and enter the correct address. For this application, you need not enter any device address for Device 6 (telecommunication controller).

When all of the necessary device addresses are correct, touch EXECUTE to save the addresses and continue on to the Primary Program Selection Menu screen, Screen 5.

7. The IDEAS Main Menu screen, Screen 5, allows you to proceed to any of the functional components of IDEAS by touching the appropriate FN key.

```
IDEAS System Utilities - Primary Program Selection Menu Module      Release 2.1

                        FN Program Module
                      ■ 00 Security
                      • 01 System date
                      • 02 Device addresses
                      • 03 Manage system peripherals
                      • 04 New user's ID
                           -- Other Functions --
                      • 06 Data files
                      • 07 Screens
                      • 08 Programs (interactive)
                      • 09 Programs (batch)
                      • 11 Reports
                      • 12 Menus
                      • 13 START programs
                      • 14 Other utilities
                      • 15 Applications

SEP 27 83                                                          Term #  2
Mem 36.00                   Global Selected                        Sta. # 20

Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Screen 5. Primary Program Selection Menu Screen

To change the system date, touch FN 01 and return to Step 5. If the system bypassed the Peripheral Device Selection screen, touch FN 02 and return to Step 6 to make sure that all of the device addresses are correct.

When you create an application, the first operation that you must perform is data file creation, because everything in an IDEAS-generated application is based on the data files. After you create the data files, there is no set order in which you must proceed.

Touch FN 06 in the Primary Program Selection screen and the Data File Utility screen, Screen 6, appears.

```
IDEAS Data File Editor - Utility Selection Menu Module            Release 2.1

                         FN Program Module
                       ■ 00 Create
                       • 01 Revise
                       • 02 Document
                       • 03 Initialize
                       • 04 Document (Batch)
                       • 05 Copy a data file control file
                            -- Other functions --
                       • 07 Screens
                       • 08 Programs (interactive)
                       • 09 Programs (batch)
                       • 11 Reports
                       • 12 Menus
                       • 13 START programs
                       • 14 Other utilities
                       • 15 Applications
                       • 31 Main IDEAS menu

SEP 27 83                                                            Term #  2
Mem 36.00                   Global Selected                          Sta. # 20

Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Screen 6. Data File Utility Screen

8. You do not have to return to the IDEAS Main Menu screen to switch from one utility to another. You can evoke the other utilities directly from the Data File Utility screen.

   To create a data file, touch FN 00. The Naming New Data File screen, Screen 7, appears.

```
IDEAS Data File Editor - Access Security Control Module          Release 2.1


      Please enter the file name for the file to be created     ■■■■■■■■
```

Screen 7. Naming New Data File

9. Enter the name MAILLIST for the data file you want to create. The Data File Creation screen, Screen 8, appears after the system accepts the file name.

```
IDEAS Data File Editor - Initial File Specification Module          Release 2.1

File name    MAILLIST  File type ■  Disk /          Revision number       by
Description                                         Last revision date    /  /

Version              Edit/Document Privilege        Data file access privilege
Application          User class / ID     /          Read  user class/ID    /
Function             Password                       Write user class/ID    /

Primary file name                 Description

   List of valid file types                 Duplicate  Random   Sequential  Alternate
                                              keys     access     access      keys
   File type   Data location                 allowed   allowed    allowed    allowed
1 Reference   None, keys only                  No       Yes        Yes         No
2 Primary     Data segments in file            No       Yes        Yes         Yes
3 Primary     Data segments in file            Yes      Yes        Yes         Yes
4 Primary     Key index element                No       Yes        Yes         No
5 Alternate   Associated primary file          No       Yes        Yes         No
6 Alternate   Associated primary file          Yes      Yes        Yes         No
7 Alternate   Associated primary file          Yes       No        Yes         No
```

Screen 8. Data File Creation Screen

10. The Data File Creation screen appears with the cursor under the entry for file type. There are seven valid file types, as indicated in the box in the lower half of the screen. These are all explained in detail in Section 3.2 of the IDEAS Release 2 Manual.

    a. Enter 2 as the file type. This indicates that this file is a primary file consisting of both keys and data that allows no duplicate keys. In this case, since the key will be an identifying customer number, no two customers can have the same customer number.

    b. The cursor moves to the Disk Address entry. The system defaults the device address entered as Device 7 on the Peripheral Device Address Selection screen, the first application data file device address. If the disk address is not correct, enter the correct one. For the next entry field, Description, enter the phrase CUSTOMERS AND ADDRESSES.

c. The next three fields (Version, Application, Function) are to set up a filing scheme for an application, for sorting data files for reports, and for installing a developed application. The IDEAS utilities also use these fields for different types of batch processing. Even though you will not use these fields for this sample application, it is generally a good idea to use these fields to associate related components of an application.

   Enter 1 as the Version, AAA as the Application, and DEMO as the Function.

d. The next three fields (User Class, User ID, and Password) are security options for editing and documenting privileges. Only a user whose user class is the indicated class or higher can edit or document this data file. In the User ID field, you can indicate a particular User ID; only a user of the indicated ID can edit or document this data file. The next field is the password field; if you indicate a password here, this password must be entered any time someone attempts to edit or document this field.

   Accept the default user class 0 and leave the User ID and Password fields blank.

e. The next two fields allow you to indicate the minimum user class that will be allowed to read from this file, and a particular user who will have exclusive access to this file. Similarly, the next two fields allow you to indicate a minimum user class and User ID for writing to the file.

   Accept 0 as the minimum read user class, bypass the User ID field, accept 0 as the minimum write user class, and bypass the next User ID field.

   A message appears at the bottom of the screen allowing you to accept or modify the information shown. Touch EXECUTE to accept and the Data File Field Definition screen, Screen 9, appears.

```
IDEAS Data File Editor - Field Attribute Edit Module          Release 2.1
File   MAILLIST                                File MAILLIST field names Pg 1 of 0
       CUSTOMERS AND ADDRESSES
FN Attribute
01 Field name
02 Sub field of
03 If sub-field, starts at byte
04 Length
05 Position in record
06 Valid character type
   0=Digits            4=U/C letters
   1=Digits & signs    5=U/C & digits
   2=Digits & dec.     6=U/C, num, & punct
   3=Any numeric       7=Any character
07 Justify
08 Zero fill
09 Decimals   ( 0-7 )
10 Security user class ( 0-9 , A-F )
FN Action                FN Action
00 Add field             19 Show name page 3
EDIT  Edit Field         20 Re-arrange for
17 Show name page 1         optimum packing
18 Show name page 2      25 Delete field
Touch desired FN key in LOWER box, FN'31 = CANCEL
```

Screen 9. Data File Field Definition Screen

11. The next step is to define the contents of the data file. Do this by entering the field names, lengths, and types. To enter a field name, touch FN 00. The system supplies the default field name of FIELD001. Override the default by entering CUST # as the first field and touching RETURN. (If the field name is 8 characters, the cursor automatically moves to the next field; otherwise you must touch RETURN.) The cursor moves to the length field. Enter 5 and touch RETURN. Note that you cannot have a field length of zero (given as the default field length) since the system will not accept this as a field length.

    A message appears at the bottom of the screen allowing you to accept the field as is with the default field attributes as shown. This field is to be right-justified, and is to consist of digits only. Before accepting the field as shown, touch FN 06 and enter 0, indicating that the field is to be comprised of digits only, and touch FN 07, indicating that the field is to be right-justified. The field must be right-justified because it is the field that determines the order of printing the report. If the field is not right-justified, the order will be incorrect (11 will print before 2, 25 will print before 3, and so forth).

    Touch EXECUTE to accept the field.

12. Define the remaining fields in the same way by performing the following steps:

    a. Touch FN 00 and enter NAME (RETURN), enter 25 as the length (RETURN), touch FN 06 and enter 4 (UC letters), and accept the field.

    b. Touch FN 00 and enter ADDRESS (RETURN), enter 25 as the length (RETURN), and accept the field.

    c. Touch FN 00 and enter CITY (RETURN), enter 16 as the length (RETURN), touch FN 06 and enter 4, and accept the field.

    d. Touch FN 00 and enter STATE (RETURN), enter 2 as the length (RETURN), touch FN 06 and enter 4, and accept the field.

    e. Touch FN 00 and enter ZIP CODE, enter 5 as the length (RETURN), touch FN 06 and enter 0, touch FN 08, and accept the field. FN 08 (Zero Fill) must be Y so that a zip code with a leading zero will print on a data entry screen in its entirety; in other words, if you do not set this attribute to Y when you define the data field, the zip code 07748 will print on a screen as 7748.

    f. Touch FN 00 and enter CLASCODE, enter 1 as the length (RETURN), touch FN 06 and enter 0, and accept the field.

    Once you have entered information in the seven fields, your screen should look like Screen 10. If it does not, touch EDIT to edit a field, FN 25 to delete a field, or FN 00 to add a field. Touch EXECUTE to accept the file and Screen 11, the Key Field Selection screen, appears.

```
IDEAS Data File Editor - Field Attribute Edit Module           Release 2.1
File    MAILLIST                                 File MAILLIST field names Pg 1 of 1
        CUSTOMERS AND ADDRESSES                  ADDRESS
FN Attribute                                     CITY
01 Field name                                    CLASCODE
02 Sub field of                                  CUST #
03 If sub-field, starts at byte                  NAME
04 Length                                        STATE
05 Position in record                            ZIP CODE
06 Valid character type
   0=Digits            4=U/C letters
   1=Digits & signs    5=U/C & digits
   2=Digits & dec.     6=U/C, num, & punct
   3=Any numeric       7=Any character
07 Justify
08 Zero fill
09 Decimals   ( 0-7 )
10 Security user class ( 0-9 , A-F )
FN Action             FN Action
00 Add field          19 Show name page 3
EDIT  Edit Field      20 Re-arrange for
17 Show name page 1      optimum packing
18 Show name page 2   25 Delete field
Touch desired FN key in LOWER box, FN'31 = CANCEL, EXEC = accept file as shown =
```

Screen 10. Complete MAILLIST Field Definition Screen.

```
IDEAS Data File Editor - Key Field Selection Module            Release 2.1
                                                 File MAILLIST field names Pg 1 of 1
File    MAILLIST                                 ADDRESS
        CUSTOMERS AND ADDRESSES                  CITY
                                                 CLASCODE
Key may be composed of up to 5 fields            CUST #
contained in the primary file's record.          NAME
Fields need not be contiguous in record          STATE
Sort order for each key component field          ZIP CODE
may be ascending (+) or descending (-).
NOTE: For Type 1 files, Field order is
determined by field attributes.

          FN  Field name  +/-
          01  ========
          02
          03
          04
          05

17 Display available field names page 1
18 Display available field names page 2
19 Display available field names page 3
```

Screen 11. Key Field Selection Screen

13. When the Key Field Selection screen appears, enter CUST # (RETURN). A plus sign (+) appears as the default in the next field, indicating that the key is to be in ascending sort order. Touch the RETURN key to accept the default. This is the only field that will comprise the key for MAILLIST. As you can see, the key could be composed of up to five noncontiguous fields.

A message appears at the bottom of the screen indicating that you can accept the key field selection, establish additional key fields, or view the field names that could not fit on this screen (if you have defined more than 84 fields). Touch EXECUTE and the Disk Space Allocation screen, Screen 12, appears.

```
IDEAS Data File Editor - Disk Space Allocation Module              Release 2.1

File  MAILLIST     CUSTOMERS AND ADDRESSES          'FN  Operation
                                                    '00  Change # of volumes*
File Type 2                  Number of volumes 1    '01  Edit volume # 1**
(Note: Only file types 2 & 3 may be multi-volume.)  '02  Edit volume # 2**
                                                    '03  Edit volume # 3**
Records Specified ########   No. provided           '04  Edit volume # 4**
                                                    '05  Edit volume # 5**
                    Sectors    Minimum    Sectors   '06  Edit volume # 6**
Vol    Disk       Available   Required   Allocated  '07  Edit volume # 7**
 1     /                                            '08  Edit volume # 8**
 2     /                                            '15  Change # of records
 3     /                                            '16  Change vol 1 address
 4     /                                            '31  Cancel
 5     /                                            * File types 2 & 3 only
 6     /                                            **Applicable only to the
 7     /                                              no. of volumes selected
 8     /                                              & only to multi-volume
 Total sectors    ______     ______     ______      Required must = Allocated
```

Screen 12. Disk Space Allocation Screen

14. Enter 500 as the number of records desired in the file and touch the RETURN key. The system displays the number of records actually provided, usually 5 percent more than the number desired. The disk address specified for this file, the number of sectors available on that disk, the number of sectors required by the data file you are currently defining, and the number of sectors the system is able to allocate also appear on the screen.

A message at the bottom of the screen allows you to accept or edit the information shown. Touch EXECUTE to accept and the Data File Initialization screen, Screen 13, appears.

```
IDEAS Data File Editor - File Initialization Module                Release 2.1

        FN  File type      Name       Description
        00  Primary    2   MAILLIST   CUSTOMERS AND ADDRESSES
        01  Alternate
        02  Alternate
        03  Alternate
        04  Alternate
        05  Alternate
        06  Alternate
        07  Alternate
        08  Alternate
        09  Alternate
        10  Alternate
        11  Alternate
        12  Alternate
        13  Alternate
        14  Alternate
        15  Alternate
        16  Alternate
        17  Initialize all files listed above, primary & alternates

        Now initializing file           Processing is     % complete
Touch EXECUTE to initialize MAILLIST, or CANCEL to abort
```

Screen 13. Data File Initialization Screen

15. The Data File Initialization module creates blank records, blank keys, and corresponding pointers to the blank records. Touch EXECUTE to initialize MAILLIST. After you initialize the file, the Data File Utility Menu screen (Screen 6) appears.

16. The next step is to document the data file. Touch FN 02 and a screen appears that allows you to enter the name of the data file you want to document. The data file name MAILLIST should already be displayed as a default; IDEAS keeps track of the last file it worked with within each utility. If MAILLIST is not displayed, enter it. Touch EXECUTE and the Data File Documentation screen, Screen 14, appears.

```
IDEAS Data File Editor - File Documentation Module                    Release 2.1
File MAILLIST T=2 Rev#     0 092783 bySJL   File MAILLIST field names Pg 1 of 1
Desc CUSTOMERS AND ADDRESSES                ADDRESS
Version 1 Application AAA Function DEMO     CITY
                                            CLASCODE
Security: Read  user class 0 user ID        CUST #
          Write user class 0 user ID        NAME
E/D password          u/cl 0 user ID        STATE
                                            ZIP CODE
Records:      Specified      500   0.00%
    0         Provided       532   0.00%
  7 fields    L 0079 N 0005 A 0000 P  78

Key fld  Pos Len S   Primary    MAILLIST
CUST #     1   5 +   Alternate key files




Packed key len    3
Pointer           2
Data in KIE       0
Total KIE len     5
Touch EXECUTE to document, FN'1-FN'3 to display field name pages, or CANCEL
```

Screen 14. Data File Documentation Screen

17. The Data File Documentation screen allows you to review the data file information on the screen before you print the documentation. Touch EXECUTE and the documentation is printed at whatever printer was specified as Device 1 in the the Device Address Selection screen. If this printer is not selected, a message appears on Line 24 indicating this. The documentation appears on the following pages.

Revision number 1 Version 1 User class 0
Last revised 09/27/56 Application code AAA User ID code
Last revised by SJL Function code DEMO Password

Read user class = 0 User ID code = Write user class = 0 User ID code =

KEY FILE	TYPE	FIELD	FIELD	FIELD	FIELD	FIELD	KEY	PACK	PTR	DATA	KIE	SORT ARRAY
MAILLIST	2	+CUST #					5	3	2	0	5	18

Record statistics: Numeric = 0005 Records currently on file = 0
No. of fields = 7 Upper case = 0000 0.00% full as specified @ 500
Record length = 0079 Packed length = 78 0.00% full as provided @ 532

Key to field attribute description column headings:

FIELD = Field name
POS = Position in record of first character in field
LEN = Length of field
T = Type of field as follows: 0 = Digits only 4 = Upper case letterd
1 = Digits & signs 5 = U/C letters & digits
2 = Digits & dec pt 6 = U/C numerics & punct
3 = Any numeric 7 = Any character
J = Justification (blank if left justified, 'R' if right)
F = Fill character for right-justified numerics (zero or blank)
D = Number of decimal specified decimal places (for numeric fields)
U = User code - specifies the minimum user level for access to the field
SUB OF = If a sub-field, the name of the parent field
POS = If a sub-field, the starting position within the parent field

Field attributes in alphabetic order

FIELD	POS	LEN	T	J	F	D	U	SUB OF	POS
ADDRESS	31	25	7				0		
CITY	56	16	4				0		
CLASCODE	79	1	0				0		
CUST #	1	5	0	R			0		
NAME	6	25	7				0		
STATE	72	2	4				0		
ZIP CODE	74	5	0				0		

Field attributes by position in record

FIELD	POS	LEN	T	J	F	D	U	SUB OF	POS
CUST #	1	5	0	R			0		
NAME	6	25	7				0		
ADDRESS	31	25	7				0		
CITY	56	16	4				0		
STATE	72	2	4				0		
ZIP CODE	74	5	0				0		
CLASCODE	79	1	0				0		

18. After it documents the data file, the system returns to the Data File Utility menu. Once you create the data file, the order in which you create the rest of the application does not matter. The next logical sequence, however, is the creation of a screen mask for the data entry module.

Touch FN 07 and the Screen Mask Utility menu, Screen 15, appears.

```
IDEAS Screen Mask Editor - Utility Selection Menu Module          Release 2.1

                            FN Program Module
                          ■ 00 Create
                          • 01 Revise
                          • 02 Document
                          • 03 Extra screen utilities
                          • 04 Document (batch)
                               -- Other Functions --
                          • 06 Data files
                          • 08 Programs (interactive)
                          • 09 Programs (batch)
                          • 11 Reports
                          • 12 Menus
                          • 13 START programs
                          • 14 Other utilities
                          • 15 Applications
                          • 31 Main IDEAS menu
SEP 27 83                                                          Term #  2
Mem 36.00                    Global Selected                       Sta. # 20
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Screen 15. Screen Mask Utility Menu

19. Touch FN 00 to load the Screen Mask Creation module and enter the file name for the screen mask you want to create (MAILSCRN). The Screen Mask Specification screen, Screen 16, appears.

```
IDEAS Screen Mask Editor - Screen Mask Specification Module          Release 2.1

Screen name MAILSCRN                                 Last revision no.        by SJL
Description ********************************         Date of last revision 09/27/83

Version              Edit & document privilege        HELP screen (if any)
Application          User class     User ID           FN keys to be trapped
Function             Password                         (up to 8 may be used)

       Data files associated with screen                 General screen attributes
FN     File    Description                           FN Attribute
01                                                   08 This is a sub screen          N
02                                                   09 Skip ahead keys enabled       Y
03                                                   10 Skip back keys enabled        Y
04                                                   11 Cancel key enabled            Y
05                                                   12 Clear work buffer @ load      Y
06                                                   13 Display all fields @ load     N
07                                                   14 First line is bright          Y

Minimum size of work buffer is      1750 bytes      Number of fields on screen      0
Minimum size of record buffer is     256 bytes      Number of boxes on screen       0
```

Screen 16. Screen Mask Specification Screen

20. The Screen Mask Specification screen appears with the cursor under the Description field. Enter MAILING LIST OF CUSTOMERS. Enter 1 as the Version, AAA as the Application, and DEMO as the Function. Accept the default of zero as the user class (there will be no security feature for this exercise). Bypass the next two fields (User ID, Password).

    The next screen asks for the name of a help screen. A help screen is a screen that contains only text. An operator can evoke a help screen from its associated screen by touching FN 15. Enter MAILHELP as the Help screen. You will define this screen later.

    Bypass the FN keys to be trapped. You will not use this feature in this application.

    A message appears at the bottom of the screen indicating that you can edit the fields you just entered, indicate which data files are to be associated with the screen, and change the default screen attributes. Touch FN 01 and enter the data file name MAILLIST as File 1, the only data file associated with this screen. Touch EXECUTE to accept the screen attributes.

21. A screen lined with dots appears with the message "1 1 FN'0 = Edit Field FN'16 = Edit Box FN'31 = Cancel EXEC = Save" displayed on Line 24. The numbers "1 1" indicate the current row and column position of the cursor.

At the top of the screen, the message "The key field for data file MAILLIST is provided below for your convenience" appears. On Line 3, the field name CUST # appears (the key field for MAILLIST, the file you defined as File 1 on the previous screen). On Line 4, pseudoblanks indicate that the field CUST # was already defined for this screen. The key field for File 1 always appears automatically on a screen being defined.

Before you begin to define the screen, eliminate this explanatory text and move the field to a more appropriate position. Touch FN 08 to eliminate the top line of text. Then touch FN 05 twice to move the cursor to Line 3, and touch FN 08 again to eliminate the field name. You will enter a less abbreviated description of the field later.

To move the defined field, touch FN 05 once more to move the cursor to Line 4, then touch FN 21 to move the cursor and the field down one more row. (Using the various function keys at screen definition is described in detail in Subsection 4.1.2 of The IDEAS Release 2 User Manual.) Then touch FN 26 22 times to position the field in Column 23.

You can now enter any text and boxes you wish to appear on the screen. You can type in all of the text and then define the fields, or you can define each field immediately after you type in the prompt for that field on the screen. Before you define the remaining fields, define the entire screen mask. At the specified row and column, type in the following information, using the two numbers in the bottom left corner of the screen to determine cursor position. You can move the cursor around the screen with the cursor keys along the top of the keyboard (FN 05, FN 06, and FN 11 through FN 14). You can also position the cursor with the RETURN key and the Space Bar, but the RETURN key always moves the cursor to column one, and the Space Bar erases screen text.

Row	Col	Mask Information
1	1	CUSTOMER MAINTENANCE SCREEN
5	6	CUSTOMER NUMBER:
7	17	NAME:
8	14	ADDRESS:
9	2	CITY/STATE/ZIP CODE:
12	11	CLASS CODE:
18	23	PRESS FN 15 FOR AN EXAMPLE OF A HELP SCREEN

22. When you define a screen, you can use the Edit Box function to box off sections of the screen or to underline text on the screen. For this exercise, create one box and one underline.

To box the address, move the cursor to Row 7, Column 22. Touch FN 16 to change the message along the bottom of the screen. The default values for the box edit row and column indicate the cursor position at the time you touch FN 16. Accept 7 and 22 as the positioning. For the DEP (depth of box) field, enter 03 because the address you want to box is three lines deep. For the WID (width of box) field, the default value indicates how wide the box must be to continue to the right edge of the screen. Enter 26 as the width because you want the box to surround a 25-character name. (The left and right edges of a box run through the middle of a character space on the terminal screen.)

To underline, create a box with a depth of zero. To underline the title of this screen, move the cursor to Row 2, Column 1 (one row beneath the text you want to underline). Touch FN 16, bypass the row, column, and depth entry (depth should have a default value of zero), and enter 27 as the width.

Your screen should now look like the Screen Mask Creation screen, Screen 17. At this point, you should note that you can use FN 21 and FN 22 (the SHIFT Up Arrow and the SHIFT Down Arrow) to move entire lines of text up or down the screen as long as the text you are moving doesn't run into an existing line of text. After you define your fields (Steps 23 through 25), experiment with these keys and note how you can use them to move fields along with associated screen text (you did this in Step 21). You can use FN 09 and FN 10 (the INSERT and DELETE keys) to move text across the screen, but you must shift these keys, as with FN 05 and FN 06, in order to move fields along with text. Other function keys that are operational during screen mask text editing are described in Subsection 4.1.2 of The IDEAS Release 2 User Manual.

```
CUSTOMER.MAINTENANCE.SCREEN.............................................

.....CUSTOMER.NUMBER:.■■■■■.

................NAME: [                         ]
.............ADDRESS: [                         ]
.CITY/STATE/ZIP.CODE: [                         ]

..........CLASS.CODE:

......................PRESS.FN.15.FOR.AN.EXAMPLE.OF.A.HELP.SCREEN..............

[2] [1] | FN'0 = Edit Field | FN'16 = Edit Box | FN'31 = Cancel | EXEC = Save |
```

Screen 17. Screen Mask Creation Screen

23. To define a field, position the cursor at the row and column where you want the field to start and touch FN 00. The Field Definition screen, Screen 18, appears. On this screen, you can define the individual field parameters to be associated with the particular field.

Since the field CUST # was defined automatically for this screen (and should be positioned at Row 5, Column 23, as described in Step 21), the first field you will define is NAME. Using the cursor keys, position the cursor at Row 7, Col 23. Touch FN 00 and the Field Definition screen, Screen 18, appears. Associated data files, as defined in the Screen Mask Specification screen (Screen 16), appear on this screen under FN 00.

```
IDEAS Screen Mask MAILSCRN - Field Attribute Specifications          Release 2.1

  FN ATTRIBUTE                                 FN ATTRIBUTE
  00 File number  0                            07 Right Justify                 N
     Data files   4=                           08 Zero fill                     N
     1=MAILLIST   5=                           09 Decimals                      0
     2=           6=                           10 Display option                2
     3=           7=                              0=Not displayed  2=Bright
  01 Field                 FIELD002               1=Normal         3=Blink
     Field number                 2            11 Keyboard entry                Y
  02 Row                          7            12 Required field                N
  03 Column                      23            13 Full if present               N
  04 Length                       0            14 EXEC required                 N
  05 Position                    80            15 Non-bypassable                N
  06 Character Type               7            16 Number of edits               0

  Valid Character Types                        FN Processing Option
  0=Digits          5=U/C & Digits             16 Access edit specifications
  1=Digits + -      6=U/C num punct            24 Copy all edits
  2=Digits & .      7=Any Character            25 Delete this field
  3=Any Numeric     8=Yy1=1,NnO=0              26 Insert as new field
  4=U/C letters     9=EXEC EDIT FN'            31 Cancel field edit
                                               FN (TAB) Display names
Touch the FN key corresponding to the desired operation or EXECUTE to accept
```

Screen 18. Field Definition Screen

24. Customer name (NAME) is part of the associated data file MAILLIST, as are all of the fields on this screen. Touch FN 00 and enter 1 as the file number. This causes MAILLIST to appear as the file name. Enter the field name NAME (RETURN) as Field Name (Field Name Number 2, since CUST # was Field 1). The system extracts the field length, position in record, character type for this particular field, and whether the field is to be zero-filled or not (FN 08) from the data file. The row and column position default from the position on the screen mask from which you evoked this screen.

    Touch EXECUTE to accept the default screen field options. The system returns to the screen mask with pseudoblanks for the location of the defined field. (On the documentation, pseudoblanks are represented by the # sign.)

25. Define the remaining fields in the same way. Position the cursor at the following locations. After positioning the cursor, touch FN 00 to define a field, then touch FN 00 again to indicate an associated data file. In all cases, enter the file number 1 for MAILLIST, then enter the appropriate field name. The field names and, for STATE, the function key to press to modify the default parameters, are:

Row	Col	Field Name	Modifications (if any)
8	23	ADDRESS (RETURN)	
9	23	CITY (RETURN)	
9	40	STATE (RETURN)	FN 13 (full if present)
9	43	ZIP CODE	FN 13 (full if present)
12	23	CLASCODE	

If you want to see the parameters that defaulted for the key field CUST #, move the cursor to any position with the CUST # field (Row 5, Columns 23 through 27) and touch FN 00. Note that the field was automatically defined as nonbypassable, required, and bright intensity. After viewing the parameters, touch EXECUTE to return to the Screen Mask Creation screen.

26. The screen you are currently defining includes a 1-byte field in the bottom right corner of the screen that will be used for processing during data entry. You can only use the space in the corner of the screen to define a field that accepts nothing but function keys as valid entries. To define this field, Touch FN 04, then touch FN 05 to move the cursor to Row 24, Column 80, then touch FN 00 to define the field.

    Touch FN 01 and enter ENDFIELD, then accept the field definition screen.

27. To create a data entry program using the screen mask and its fields, you must use the Screen Mask utilities to indicate what field processing is associated with each field. All of the programming logic associated with the data entry program is implemented through the field processing. Field processing is stored in an Edit Specification file and later used by the Program Generation utility.

    To avoid redefining frequently-used field processing sequences, IDEAS allows you to recall any previously defined set of field edits and associate them with a current field, at which point you can modify them further. The IDEAS Release 2 utilities come complete with a few standard field edits involving simple data entry. In this exercise, you will recall and modify these library edits.

    All of the library edits you will use come from the library screen IDS2sACS. This screen includes field processing from a simple add, change, and delete a record program. In this program, all checking for duplicate keys occurs when the record is complete and is being saved, but not at the time you enter the key value.

    In order to copy the processing from a library screen, you must retrieve the screen from archive so that its Edit Specifications are contained in the Edit Specification file. If the library screen IDS2sACS was already retrieved from archive (either at installation or during a previous run-through of this exercise), continue with Step 28. If it was not, perform the following steps:

a. Touch EXECUTE to save what you defined for the current screen. The Screen Utilities screen appear.

b. Touch FN 03 to evoke the Supplementary Screen utilities. Touch FN 00 to evoke the Screen Mask Filing screen. Refer to Subsection 4.2.3 of <u>The IDEAS Release 2 User Manual</u> for information on using this utility. Use Option 3 to retrieve the screen IDS2sACS from the disk address where the IDEAS Utilities reside.

c. After indicating that filing options are complete, the Menu Utilities screen should reappear. Touch FN 01, enter MAILSCRN as the screen you want to revise, accept the Screen Mask Specification screen to return to the Screen Mask Creation screen, and continue with Step 28.

28. You can specify many processing sequences for each field on the screen. When copying edits from a library screen (or from any previously defined screen), you can copy individual sequences from each field, in the manner described in Steps 28 through 41, or all of the edits from any particular field can be copied using the Copy All Edits option (FN 25) on the Field Definition screen (Field 18). Although for this example you will, in fact, copy all of the edits from two fields, you will copy them one sequence at a time so that you can see how to define each separate operation sequence.

    All field processing for this exercise is performed on the first and last fields on the screen. Move the cursor to Row 5, Column 23 (the customer number field) and touch FN 00 to recall the Field Definition screen.

    Touch FN 16 and the Edit Operation Sequence Selection screen, Screen 19, appears.

```
IDEAS Screen Mask Editor        Edit Operation Sequence Selection  Release 2.1
Note: Edit specification file is 77 % full.  511 records may be added.
For screens only, 1-32 are pre-entry and 33-64 are post-entry.  For report and
batch there is is no pre/post distinction.  To select item to be edited, move
the cursor to the desired number using FN 5,6,12,&13 (or N,S,E,W), then "EDIT"
          Pre-entry Operations                      Post-entry Operations
No Description      No Description          No Description      No Description
01 - available -    17 - available -        33 - available -    49 - available -
02 - available -    18 - available -        34 - available -    50 - available -
03 - available -    19 - available -        35 - available -    51 - available -
04 - available -    20 - available -        36 - available -    52 - available -
05 - available -    21 - available -        37 - available -    53 - available -
06 - available -    22 - available -        38 - available -    54 - available -
07 - available -    23 - available -        39 - available -    55 - available -
08 - available -    24 - available -        40 - available -    56 - available -
09 - available -    25 - available -        41 - available -    57 - available -
10 - available -    26 - available -        42 - available -    58 - available -
11 - available -    27 - available -        43 - available -    59 - available -
12 - available -    28 - available -        44 - available -    60 - available -
13 - available -    29 - available -        45 - available -    61 - available -
14 - available -    30 - available -        46 - available -    62 - available -
15 - available -    31 - available -        47 - available -    63 - available -
16 - available -    32 - available -        48 - available -    64 - available -
Attention:      Touch EXECUTE to return to field attribute specifications
```

Screen 19. Edit Operation Selection Screen

This screen allows you to define a series of operations to be performed before and after entry to the specified field.

29. The first pre-entry processing operation should be highlighted. Touch EDIT to evoke the Edit Operation Type Selection screen, Screen 20.

```
IDEAS Screen Mask Editor        Edit Operation Type Selection        Release 2.1

        FN Type of operation
        -- ---------------------------------------------------------

        00 Perform pass/fail actions only using last P/F condition

        01 Set field(s) equal to field(s) and/or constant(s)

        02 Read a record from a data file

        03 Perform logical test(s)

        04 Math calculation(s)

        05 Range test(s)

        06 Table look-up or table look-up and replace

        07 User-supplied process

Attention:      Select operation using FN'00-07 above or FN'31 to cancel
```

Screen 20. Edit Operation Type Selection Screen

As the screen indicates, you can define several different types of operations. These options are explained in Section 5.3 of The IDEAS Release 2 User Manual. You will not use most of these options in this exercise.

30. Touch FN 00. The Perform Pass/Fail Action screen (Screen 21) appears.

```
IDEAS Screen Mask Editor          No-op - Perform Pass/Fails Only     Release 2.1

Screen      MAILSCRN    When a no-op is used, there is no operation performed to
Field       CUST #      establish a new pass/fail condition.  Pass/fail actions
Sequence number 01      specified for this edit will be performed based on the
Conditional Y/N? N      currently existing pass/fail condition established by a
Only if SYSFLAG =       prior entry, edit, or operation only.

A no-op may be used in a number of situations:

1. If used immediately following a keyboard entry, the pass/fail condition is
   set by the validity of the entry.  For example, entry of an invalid date,
   when a date edit is specified will cause a "fail" condition to exist.

2. If used immediately following another edit type (ie. range, table, etc.),
   it extends that edit's number of pass/fail actions beyond the normal 50
   because the pass/fail is not changed, and this becomes merely an extension.

3. If used merely to specify certain activities, the P/F code should be "B"
   (both) because the currently existing pass/fail condition is indeterminate
   since a no-op does not set the pass/fail condition.
```

Screen 21. Perform Pass/Fail Action Screen

When Screen 21 appears, the cursor is by the question "Conditional Y/N?". To use edits from one of the library screens (described in Step 27), you must backspace the cursor to the Screen field.

Touch the backspace cursor (FN 13) three times (or FN 14 once) and enter IDS2sACS as the screen; this indicates that you are using one of the operation sequences you already defined. Then enter FIELD001 as the field; this indicates which field you are choosing an operation sequence from. Indicate 03 as the sequence number; this is the number of the sequence, as specified in the Edit Operation Selection screen (Screen 19), that you are copying. MAILSCRN and CUST # should reappear in the fields you just changed and a message should appear at the bottom of the screen indicating that the sequence you just specified was found.

Touch EXECUTE twice, once to accept the recalled operation sequence and once to accept the screen.

31. The Pass/Fail Action Specification screen (Screen 22) appears with one action already specified. This is the operation sequence you just copied from the library screen.

```
IDEAS Screen Mask Editor          Pass/Fail Action Specifications  Release 2.1

MAILSCRN CUST #    (01-0)   Touch EXECUTE to accept, or position cursor at the
                            desired P/F number using FN 5,6,12,&13 (or N,E,S,W)
   P=Pass, F=Fail, B=Both   then touch EDIT, DELETE (FN'9), or INSERT (FN'10)
    Action specification
     Operand1 Operand2      17    available           34    available
01       available          18    available           35    available
02       available          19    available           36    available
03 B H DISPMSG1             20    available           37    available
04       available          21    available           38    available
05       available          22    available           39    available
06       available          23    available           40    available
07       available          24    available           41    available
08       available          25    available           42    available
09       available          26    available           43    available
10       available          27    available           44    available
11       available          28    available           45    available
12       available          29    available           46    available
13       available          30    available           47    available
14       available          31    available           48    available
15       available          32    available           49    available
16       available          33    available           50    available
EXEC = Accept, EDIT = Modify, FN'0 = Op spec, '16 = Messages, '31 = Cancel
```

Screen 22. Pass/Fail Action Specification Screen

The action indicated on the screen specifies that as the pre-entry processing, the message defined as Message 1 will appear on Line 24. Touch FN 16 and view the contents of Message 1.

Change the pre-entry message. To do this, touch FN 01, then touch EDIT to change Message 1 to "Enter key field for desired record or CANCEL (FN 31)". Accept the Message screen.

If you wish to see a summary of what the different pass/fail options are, touch the EDIT key. The EDIT key allows you to define additional pass/fails actions on this screen. The 26 pass/fail options appear on the screen after you touch EDIT. A complete explanation of the pass/fail format and the pass/fail options is given in Sections 5.4 and 5.5 of <u>The IDEAS Release 2 User Manual</u>. Touch the CANCEL key to leave this display. You will not define any additional actions on this screen.

32. Accept the Pass/Fail screen to return to the Edit Operation Sequence Selection screen. Use FN 12 to move the highlight to Operation 33 for post-entry processing and touch EDIT.

    When the Edit Operation Type Selection screen (Screen 20) appears, touch FN 02 and the Read Record screen (Screen 23) appears.

```
IDEAS Screen Mask Editor          Read a record from a data file      Release 2.1

Screen     MAILSCRN        File access options
Field      CUST #        0  No Read, Set Field Only (No Optional Read Allwd)
SEQUENCE         33      1  Random record with key specified
Conditional Y/N? N       2  First logical record with key >= that specified
Only if SYSFLAG =        3  Next logical record

Data file                Options 1 & 2 require a key specification below :
Access option =                 F/C Field/Constant                      Len
Only if SYSFLAG =        Key spec
Else use option
Error message on
Record read mode
Inquiry=I,Update=U

 If record is found set  Option 3 assumes the following :
 work buffer area for
 field/file:                1. Option 2 Find First Logical record has been
 equal to record buffer        performed on the file prior to this read
 field/file:                2. No sequential record read has been performed
 If record is found            on other files since last option 2 or 3 read.
 set System Flag =
```

Screen 23. Read Record Screen

This screen defines the method of reading a record from a data file according to one of several options, as indicated on this screen and described in detail in Section 5.3 of The IDEAS Release 2 User Manual. You can make the record read conditional and use it to fill either one or several screen fields.

33. Touch FN 14 and enter IDS2sACS, FIELD001, and Sequence 35. A message appears that indicates that the sequence was found, and the remaining fields on the screen fill with default values taken from the library sequence. Touch EXECUTE to accept the sequence.

    All of the default values except for length of the key specification are appropriate for the application you are developing. Touch EDIT to step through each of these defaults.

    a. Accept the first three fields, which are: Y for conditional, if SYSFLAGG=N. System Flag G, which is set during post-entry processing for the last field on the screen, indicates that a record was displayed and will now be edited.

b. The name of the data file to be read is FILE # 1. This refers to the associated file specified on the Screen Mask Specification Screen (Screen 16) as File 1. You want to read from the File MAILLIST, which is File 1, so FILE # 1 is appropriate as the data file to be read (You could also specify MAILLIST here.) When you accept the name of the data file, the length of the key specifications (on the right side of the screen) automatically changes from 10 to 5.

   The access option is one. The next field (SYSFLAG_) should remain blank, since the access option is not conditional. Accept N to indicate that system error messages are not on.

c. The record read mode is U. This means that you can add or edit records. The next two fields (Set screen field ____ equal to field ______) default to FILE # 1, the file specified as the data file. This means that if the record is found, all of the screen fields that correspond to data fields in the record are set to the fields of the record found.

d. The next two fields read: Set System Flag D = Y. This flag helps you determine if the record you are trying to read is in use by another partition. If a record is in use (protected), the record read fails, but this flag is still set.

e. The next three fields determine the key field for reading the record. These fields default to F (the key is a field), CUST # (the field currently being edited), and Length 5. The length originally appeared as 10, but automatically changed to 5 when you stepped through the data file field.

34. Touch EXECUTE to accept the screen and the Perform Pass/Fail Action screen appears as shown in Screen 24.

```
IDEAS Screen Mask Editor            Pass/Fail Action Specifications  Release 2.1

MAILSCRN CUST #    (33-2)    Touch EXECUTE to accept, or position cursor at the
                             desired P/F number using FN 5,6,12,&13 (or N,E,S,W)
   P=Pass, F=Fail, B=Both    then touch EDIT, DELETE (FN'9), or INSERT (FN'10)
   Action specification
     Operand1 Operand2
01       available           17       available          34       available
02       available           18       available          35       available
03 F H DISPERR1 IFFLGDON     19       available          36       available
04 F Q CUST #   IFFLGDON     20       available          37       available
05       available           21       available          38       available
06 P U CUST #                22       available          39       available
07       available           23       available          40       available
08 P Q LAST                  24       available          41       available
09       available           25       available          42       available
10       available           26       available          43       available
11       available           27       available          44       available
12       available           28       available          45       available
13       available           29       available          46       available
14       available           30       available          47       available
15       available           31       available          48       available
16       available           32       available          49       available
                             33       available          50       available
EXEC = Accept, EDIT = Modify, FN'0 = Op spec, '16 = Messages, '31 = Cancel
```

Screen 24. Pass/Fail Action Specification Screen

The actions indicated on the screen are the post-entry options for the Library screen; you must modify one field to accommodate the screen you are defining. Using the space bar or the down arrow, move the highlight to action 08 and touch EDIT. Accept P, accept Q, and change the field LAST to ENDFIELD. Leave the second operand blank, and accept the pass/fail action.

The options now indicated on this screen perform the following actions:

03 - If the record read fails, and Flag D is on (indicating that a record was found), sound the audio alarm and display Message 1 on Line 24. Message 1, which you can view by touching FN 16, reads "Record is in use by another partition."

04 - If the record read fails, and Flag D is on, branch to the field CUST # (again) and perform any pre-entry processing for that field. In other words, start the screen over.

06 - If the record read is successful, display all fields on the screen starting at CUST #.

08 - If the record read is successful, branch to the field ENDFIELD.

35. Accept the complete Pass/Fail Specification screen.

Accept the Operation Sequence screen then accept the Field Attribute screen (which should indicate 2 as the number of edits) to return to the Screen Definition screen.

36. Touch FN 04 and FN 05 to move the cursor to the last field.

Touch FN 00 to evoke the Field Attribute Specification screen. Then touch FN 16 to indicate that you will define field processing.

The Edit Operation Sequence Selection screen appears with the first pre-entry operation highlighted. Touch EDIT, then touch FN 00 to indicate which operations you will perform on this field before field entry.

37. When the next screen (The Perform Pass/Fail Action screen) appears, touch FN 14, enter IDS2sACS as the screen, LAST as the field, and 04 as the sequence number. Accept the library specification.

Accept the screen and the Pass/Fail Operation screen appears. There is only one pre-entry action, to display Message 1 on Line 24. Touch FN 16 to view Message 1.

Accept the Pass/Fail screen to return to the Operation Sequence screen.

38. Move the cursor to Operation 33 and touch EDIT. When the Operation Type Selection screen appears, touch FN 00.

When the next screen appears, touch FN 14, enter IDS2sACS as the screen, LAST as the field, and 39 as the sequence number. Accept the library specification.

Accept the screen and the Pass/Fail Operation screen appears. The pass/fail actions are already specified, as shown in Screen 25.

```
IDEAS Screen Mask Editor          Pass/Fail Action Specifications  Release 2.1

MAILSCRN ENDFIELD (33-0)   Touch EXECUTE to accept, or position cursor at the
                           desired P/F number using FN 5,6,12,&13 (or N,E,S,W)
  P=Pass, F=Fail, B=Both   then touch EDIT, DELETE (FN'9), or INSERT (FN'10)
  Action specification
    Operand1 Operand2
01       available           17 B Q FIELD001       34      available
02 B     SKIP07PF IFLFK=09   18      available     35      available
03 B     SKIP05PF IFLFK=32   19      available     36      available
04 B     SKIP02PF IFLFK=33   20      available     37      available
05 B H RINGBELL              21      available     38      available
06 B Q ENDFIELD              22      available     39      available
07 B     FLAGG=ON            23      available     40      available
08 B Q FIELD001 IFLFK=33     24      available     41      available
09 B H DISPMSG2 IFLFK=32     25      available     42      available
10 B H DISPMSG1 IFLFK=09     26      available     43      available
11 B     SKIP01PF IFLFK=32   27      available     44      available
12 B I BLANKFLD FILE # 1     28      available     45      available
13 B O FILE # 1              29      available     46      available
14 B C FLAGG=OF FLAGD=OF     30      available     47      available
15 B I BLANKFLD MAILSCRN     31      available     48      available
16 B U FIELD001              32      available     49      available
                             33      available     50      available
EXEC = Accept, EDIT = Modify, FN'0 = Op spec, '16 = Messages, '31 = Cancel
```

Screen 25. Pass/Fail Specification Screen

39. On the Pass/Fail Screen, edit the following operations in the manner described in Step 34:

    a. For Operation 08, change FIELD001 to CUST #.

    b. For Operation 16, change FIELD001 to CUST #.

    c. For operation 17, change FIELD001 to CUST #.

40. The options now indicated on this screen perform the following actions. (Note that if the function key entered to the field is 31, the operator cancels out of the screen and these post-entry actions are never performed.)

    <u>02</u> - If the function key entered to this field was 09 (Delete), skip the next seven Pass/Fail actions (go to action 10).

    <u>03</u> - If the function key entered to this field was 32 (RETURN or EXECUTE), skip the next five P/F actions (go to action 9).

    <u>04</u> - If the function key entered to this field was 33 (EDIT), skip the next two P/F actions (go to action 7)

<u>05</u> - Sound the audio alarm. Note that Actions 5 and 6 occur only if the entry made to the field was not valid (i.e., not an EXECUTE, Delete, Edit, or CANCEL).

<u>06</u> - Branch to ENDFIELD. This starts field processing over for this field, beginning with pre-entry processing.

<u>07</u> - Turn Flag G on. This occurs only on EDIT, and notes that another record should not be read as post-entry to the first field on the screen and that the message on CUST # will not appear.

<u>08</u> - If the function key entered to this field was 33 (EDIT), branch to CUST #, the first field on the screen.

<u>09</u> - If the function key entered to this field was 32 (RETURN or EXECUTE), display Message 2. Message 2 reads "Saving record to file."

<u>10</u> - If the function key entered to this field was 09 (DELETE), display Message 1. Message 1 reads "Deleting record from file."

<u>11</u> - If the function key entered to this field was 32 (RETURN or EXECUTE), skip the next P/F. (Go to Action 13.)

<u>12</u> - Copy all blanks into the screen fields associated with File 1, which in this case is MAILLIST.

<u>13</u> - Save to disk what is currently in the screen fields associated with File 1. If action 12 was performed, blanks are copied into the file, deleting it. Otherwise, this is the action that saves the entered or edited record onto disk.

<u>14</u> - Enable the skip-back keys and turn Flag G and Flag D back to their off state to prepare for the processing of the next record.

<u>15</u> - Blank out all of the screen fields, in preparation for the processing of the next record.

<u>16</u> - Display all screen fields, starting with CUST #. Since all of the fields were blanked out in action 15, this clears the screen of all fields.

<u>17</u> - Branch to the field CUST #, where data entry processing can begin again.

41. Accept the Pass/Fail screen, then accept the Operation Sequence screen, then accept the Field Attribute screen. This returns you to the Screen Definition screen.

42. When you have created the screen mask to your satisfaction and you have defined the fields properly, touch EXECUTE to save the screen and end screen mask definition. In this case, touching the RETURN key instead of the RUN or EXECUTE key does not save the screen mask. The system returns to the Screen Mask Utility screen.

43. Touch FN 02 to document MAILSCRN. A screen appears that asks what screen to document. MAILSCRN should be the default for this field because it was the last screen you worked with. Accept this default. A message appears that asks if the field edits are to be included in the documentation. Accept the default of Y(es). The Screen Mask Documentation screen, Screen 26, appears. This screen is nearly identical to the Screen Mask Specification screen, Screen 16.

```
IDEAS Screen Mask Editor - Screen Mask Documentation Module          Release 2.1

Screen name MAILSCRN                            Last revision no.    0    by SJL
Description MAILING LIST OF CUSTOMERS           Date of last revision 09/27/83

Version          1   Edit & document privilege  HELP screen (if any)   MAILHELP
Application    AAA   User class 0   User ID     FN keys to be trapped
Function      DEMO   Password                   (up to 8 may be used)

       Data files associated with screen          General screen attributes
FN   File     Description                       FN Attribute
01 MAILLIST CUSTOMERS AND ADDRESSES             08 This is a sub screen        N
02                                              09 Skip ahead keys enabled     Y
03                                              10 Skip back keys enabled      Y
04                                              11 Cancel key enabled          Y
05                                              12 Clear work buffer @ load    Y
06                                              13 Display all fields @ load   N
07                                              14 First line is bright        Y

Minimum size of work buffer is     1750 bytes   Number of fields on screen     8
Minimum size of record buffer is    256 bytes   Number of boxes on screen      2

Attention:       Touch EXECUTE to print the documentation or FN'31 to cancel   ■
```

Screen 26. Screen Mask Documentation Screen

44. Touch EXECUTE again and the system documents the screen that you just created. If a printer is not available, a message appears to indicate this. A message also appears to indicate if the documentation is being printed. The next pages show the screen documentation for MAILSCRN

```
Revision number      1        Version               1      User class       0
Last revised 09/27/56         Application code  AAA        User ID code
Last revised by   SJL         Function code     DEMO       Password

          1         2         3         4         5         6         7         8
 12345678901234567890123456789012345678901234567890123456789012345678901234567890  01
01CUSTOMER MAINTENANCE SCREEN                                                       02
02                                                                                  03
03                                                                                  04
04                                                                                  05
05    CUSTOMER NUMBER: #####                                                        06
06                                                                                  07
07               NAME: #########################                                    08
08            ADDRESS: #########################                                    09
09 CITY/STATE/ZIP CODE: ################ ## #####                                   10
10                                                                                  11
11                                                                                  12
12         CLASS CODE: #                                                            13
13                                                                                  14
14                                                                                  15
15                                                                                  16
16                                                                                  17
17                                                                                  18
18                     PRESS FN 15 FOR AN EXAMPLE OF A HELP SCREEN                  19
19                                                                                  20
20                                                                                  21
21                                                                                  22
22                                                                                  23
23                                                                                 #24
24
 12345678901234567890123456789012345678901234567890123456789012345678901234567890
          1         2         3         4         5         6         7         8
```

Data files associated with this screen	Bytes
1 MAILLIST CUSTOMERS AND ADDRESSES	0001-0079

General screen attributes	
This is a sub screen	N
Skip ahead keys enabled	Y
Skip back keys enabled	Y
Cancel key (FN'31) enabled	Y
Clear work buffer @ load	Y
Display all fields @ load	N
First line is bright	Y

```
Field attribute key:  BEG   Starting Buffer Pos.    KBD   Allow Keyboard Entry
FLD# Field Number     END   Ending buffer pos.      REQ   Required Entry
NAME Field Name      *TYPE  Character type (0-9)    FULL  Must be Full if Used
FILE Associated File  JUST  Right justified         EXEC  EXECUTE Required
ROW  Row on Screen    ZFIL  Zero filled (left)      NBPS  Non-bypassable
COL  Column on Screen DEC   Decimal places          (by skip ahead/back keys)
LEN  Length of Field **DISP Display Option (0-3)    #EDT  Number of Special Edits

    *TYPE 0=Digits      2=Digits & .  4=U/C letters  6=U/C,num,punct 8=Yyl=Y,Nn0=N
          1=Digits + -  3=Any numeric 5=U/C & digits 7=Any character 9=EXEC,EDIT,FN
   **DISP 0=No display  1=Normal      2=Bright       3=Blinking

HELP screen name if any = "MAILHELP"  # of fields =  8  Min. record buffer =  256
                                      # of boxes  =  2  Min. work buffer   = 1750
Trapped function keys (screen level): None
```

FLD #	NAME	FILE (IF ANY)	ROW	COL	LEN	POSITION BEG	END	TYPE	JUST	ZFIL	DEC	DISP	KBD	RQD	FULL	EXEC	NBPS	# EDT	FLD #
1	CUST #	MAILLIST	5	23	5	1	5	0	Y	.	0	2	Y	Y	.	.	Y	2	1
2	NAME	MAILLIST	7	23	25	6	30	7	.	.	.	2	Y	.	.	.	.	.	2
3	ADDRESS	MAILLIST	8	23	25	31	55	7	.	.	.	2	Y	.	.	.	.	.	3
4	CITY	MAILLIST	9	23	16	56	71	4	.	.	.	2	Y	.	.	.	.	.	4
5	STATE	MAILLIST	9	40	2	72	73	4	.	.	.	2	Y	.	Y	.	.	.	5
6	ZIP CODE	MAILLIST	9	43	5	74	78	0	.	Y	0	2	Y	.	Y	.	.	.	6
7	CLASCODE	MAILLIST	12	23	1	79	79	0	.	.	0	2	Y	.	.	.	.	.	7
8	ENDFIELD		24	80	1	80	80	9	.	.	.	0	Y	.	.	.	Y	2	8

Box/Line Documentation:

Type	R	C	V	H	Type	R	C	V	H
Box	7	22	3	26	Line-H	2	1	--	27

Field "CUST # " pre entry (01) pass/fails only
01 Display message below @ row 24, column 1
"Enter key field for desired record or CANCEL (FN 31)"

Field "CUST # " post entry (33) read record: Do only if flag G is OFF
Key consists of: field CUST # (Length = 10)
Read from file MAILLIST for update or delete without system error messages
Read record randomly with key specified above
Set the work buffer area for file MAILLIST equal to the record read
If record is found, turn sysflag D ON. If not, turn it OFF
01 If FAIL and sysflag D is ON display error message below @ row 24, column 1
"Record is in use by another partition"
02 If FAIL and sysflag D is ON branch to field CUST #
03 If PASS display from field CUST # to last field
04 If PASS branch to field ENDFIELD

Field "ENDFIELD" pre entry (01) pass/fails only
01 Display message below @ row 24, column 1
"'EXEC' = Accept, FN '9 = Delete, 'EDIT' = Edit, FN '31 = Cancel"

Screen "MAILSCRN" - MAILING LIST OF CUSTOMERS 09/27/56 Page 2.0001

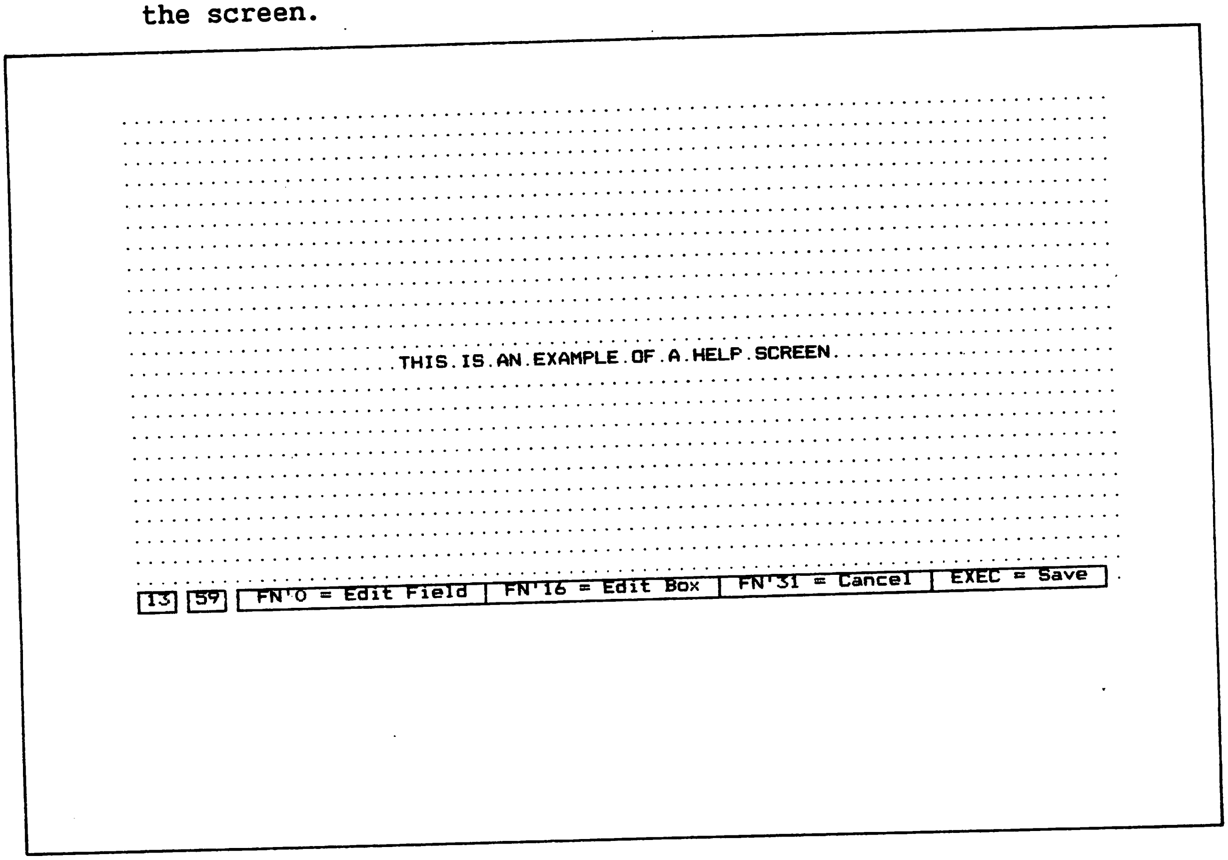
Field "ENDFIELD" post entry (33) pass/fails only
01 If last FN key is 09 skip ahead 7 P/F's
02 If last FN key is EXECUTE skip ahead 5 P/F's
03 If last FN key is EDIT skip ahead 2 P/F's
04 sound audio alarm
05 Branch to field ENDFIELD
06 Turn sysflag G ON
07 If last FN key is EDIT branch to field CUST #
08 If last FN key is EXECUTE display message below @ row 24, column 1
"Saving record to file"
09 If last FN key is 09 display message below @ row 24, column 1
"Deleting record from file"
10 If last FN key is EXECUTE skip ahead 1 P/F
11 Copy all blanks to record MAILLIST
12 Save record for file MAILLIST
13 Turn sysflag D OFF and turn sysflag G OFF and enable skip back keys
14 Copy all blanks to the work buffer
15 Display from field CUST # to last field
16 Branch to field CUST #

45. To define the help screen, touch FN 00 from the Screen Utilities menu and enter MAILHELP as the screen you want to create.

Enter HELP SCREEN FOR MAILING LIST as the description, and enter 1 as the Version, AAA as the Application, and DEMO as the Function.

Accept 0 as the User class for Edit and Document Privilege, bypass User ID and Password, bypass HELP screen, and bypass FN keys to be trapped.

Accept the screen. The system presents the Screen Definition screen. Enter the information displayed in Screen 27 (or any text you desire, as long as you do not define fields) and accept the screen.



Screen 27. Help Screen

46. Next, create the actual application programs. The first one to create is the Data Entry program itself. From the Primary Program Selection menu, touch FN 08 (labeled Programs (Interactive) on the screen) to load the Data Entry Program Utility menu, Screen 28.

```
IDEAS System Utilities - Interactive Program Generation Menu          Release 2.1

                         FN Program Module
                       ■ 00 Create
                       • 01 Revise
                       • 02 Protect
                            -- Other Functions --
                       • 06 Data files
                       • 07 Screens
                       • 09 Programs (batch)
                       • 11 Reports
                       • 12 Menus
                       • 13 START programs
                       • 14 Other utilities
                       • 15 Applications
                       • 31 Main menu

SEP 27 83                                                          Term #  2
Mem 36.00                 Global Selected                          Sta. # 20
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Screen 28. Data Entry Program Utility Menu

47. Touch FN 00 to create a data entry program. When the next screen appears, enter MAILPROG as the program to be entered. The Data Entry Program Generation screen, Screen 29, appears.

```
IDEAS Interactive Program Generator - Program Specification Module Release 2.1

Program MAILPROG - ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■  Rev #        /  /   by

Files used:    Name        Description                         Length Begin   End
Screen mask =          -
Data file 1 =          -
Data file 2 =          -
Data file 3 =          -
Data file 4 =          -
Data file 5 =          -
Data file 6 =          -
Data file 7 =          -

Version           Edit/Document Privilege   Execution Privilege   FN'0 Protect? N
Application       User class / ID 0 /       User class/ID 0/      FN'1 REMarks? N
Function          Password     "       "    Password "        "   FN'2 Mem/Load M

The Mem/Load option (FN'2) permits optimization for load   FN Program Type
time (L) or memory use (M) in the generated program code   '3 Add/change/delete
FN'3-FN'6 at right permit automatic generation of edits    '4 Add only
for the standard program types listed.                     '5 Inquiry (random)
Touch FN'15 for more information regarding these options   '6 Inquiry (seq.)
```

Screen 29. Data Entry Program Generation Screen

48. Enter MAILING LIST PROGRAM as the program description. Enter MAILSCRN as the screen mask and MAILLIST appears as the name of Data File 1. Enter 1 as the Version, AAA as the Application, and DEMO as the Function. Accept 0 as the User Class for Edit/Document Privilege, and bypass the next two fields. Accept 0 as the User Class for Execution Privilege and bypass the next two fields. Enter what you like for the next three fields: the Protect option indicates that the program can only be modified through the IDEAS utilities. The REMs options indicates whether the generated program will include internal documentation. The Mem/Load option allows you to optimize for load time or memory use in the generated program code.

    Accept the screen. A series of messages appears at the bottom of the screen indicating that the program and its individual edits are being compiled. When program generation is finished, the system returns to the Data Entry Program Utility menu.

49. The next program to create is the START program. Touch FN 13 to load the START Utilities menu. The START Utility Menu screen, Screen 30, appears.

```
IDEAS "START" Program Editor - Utility Selection Menu Module        Release 2.1

                    FN Program Module
                  ■ 00 Create
                  • 01 Revise
                  • 02 Document
                       -- Other Functions --
                  • 06 Data files
                  • 07 Screens
                  • 08 Programs (interactive)
                  • 09 Programs (batch)
                  • 11 Reports
                  • 12 Menus
                  • 14 Other utilities
                  • 15 Applications
                  • 31 Main IDEAS menu

SEP 27 83
Mem 36.00               Global Selected                       Term #  2
                                                              Sta. # 20
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Screen 30. Start Utility Menu Screen.

50. Touch FN 00 to create the START program. The next screen allows you to enter the name for the module to be created. Enter MAILSTRT. After you enter the last character, the START Program Creation screen, Screen 31, appears.

```
IDEAS "START" Module Editor - General Specifications                Release 2.1

Module name   MAILSTRT                                   Revision number          By SJL
Description   ################################          Last revision date     09/27/83

Version              Edit & document privilege         Application access privilege
Application          User class     User ID            User class     User ID code
Function             Password                          Password

Peripheral device address assignments   #08/ D21 Application data files or other
#01/ 216 Printer                        #09/ D21 Application data files or other
#02/ D23 IDEAS system utilities         #10/ D21 Application data files or other
#03/ D21 Screen & Report masks          #11/ D21 Application data files or other
#04/ D21 Application programs           #12/ D21 Application data files or other
#05/ D21 Data file control files        #13/ D21 Application data files or other
#06/     Telecommunications board       #14/ D21 Application data files or other
#07/ D21 Application data files         #15/ D21 Application data files or other

Program module to load in case of access privilege or other failure =
Program module to load under proper, normal execution circumstances =
Set @SYSBUFO=
FN'16 Work buffer size      ( 1750)   1750   FN'18 Sort buffer size (  256)     256
FN'17 Record buffer size (  256)      256    Number of primary files opened =     0
```

Screen 31. START Program Creation Screen

51. The first entry to make on the START Program Creation Screen is a description of the program. Enter MAILING LIST PROGRAM. Enter 1 as the Version, AAA as the Application, and DEMO as the Function. Accept 0 as the User Class for Edit and Document Privilege and bypass the next two fields. Accept 0 as the User Class for Application access privilege, and bypass the next two fields.

Bypass the next field, which asks for the program module to load in case of access privilege or other failures, since you have not defined an appropriate module.

In the next field, enter MAILMENU as the program to load under normal circumstances.

Bypass the next field (@SYSBUF=), which refers to a system buffer supplied by IDEAS. You will not use this feature for this application.

A message appears at the bottom of the screen indicating that you can perform a variety of actions. The device addresses should be the same as those indicated on the device address selection screen, so you do not need to change these. The work buffer, record buffer, and sort buffer size should all be sufficient for our application. (A complex application may require larger buffers; the required work buffer size is indicated on the Screen Mask Specification screen after you define all edits, and the required record and sort buffer sizes are provided automatically.) Accept the screen, and the START Module Data File Specification screen, Screen 32, appears.

```
IDEAS "START" Module Editor - Data File Specifications                  Release 2.1
Total number primary and alternate files open =        | Primary files selected
FN   File    Description                       Use Sort |
-    ------- -------------------------------   --- ---- |
             (Primary file)
00
             (Alternates key files if any)
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
Use codes S=Sequential/random R=Random only N=Not used   Sort=Sort buffer size
EXEC=Accept   'O =Add a file  FN'15=Help FN'31=Cancel
```

Screen 32 - START Module Data File Specification Screen

52. Touch FN 00 to add a file, then enter MAILLIST as the primary file to be opened. Touch EXECUTE twice to accept the file and to accept the screen and return to the START Program Creation screen.

    Before you accept the START program, make sure the peripheral device address assignments are correct. Accept the screen. The system returns to the START Utility Menu screen. You can document the START program if you wish.

53. The next step in the development of the application is to create the application menu program to be loaded from the START module you have just created. To do this, touch FN 12. The Menu Utility screen, Screen 33, appears.

```
IDEAS Application Menu Editor - Utility Selection Menu Module      Release 2.1

                         FN Program Module
                       ■ 00 Create
                       • 01 Revise
                       • 02 Document
                       • 03 Document (batch)
                            -- Other Functions --
                       • 06 Data files
                       • 07 Screens
                       • 08 Programs (interactive)
                       • 09 Programs (batch)
                       • 11 Reports
                       • 13 START programs
                       • 14 Other utilities
                       • 15 Applications
                       • 31 Main IDEAS menu

                                                                    Term #  2
SEP 27 83                     Global Selected                       Sta. # 20
Mem 36.00
Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default
```

Screen 33. Menu Utility Screen

54. Touch FN 00 to create the application menu screen and menu program. Enter MAILMENU for the menu you want to create. The Menu Definition screen, Screen 34, appears.

```
IDEAS Application Menu Editor - Menu Specification Module          Release 2.1

Menu name MAILMENU  Rev no     Revised 09/27/83 by SJL  Enter menu title below
*******************************************************************************
Version    Application      Function       User class   UID     Password
FN Program  Description (for display)        Program type (I,W,X) UC UID Password
00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16                                                               (CANCEL entry)
```

Screen 34. Menu Definition Screen

55. From this menu, you want to be able to access the data entry program. Typically, menus contain data entry programs, reports, and other menus. For this application, however, the menu will have only two entries: one for the program you just generated, and one for the report you will define later.

For the description of the menu enter CUSTOMER MAILING LIST PROGRAM. Enter 1 as the Version, AAA as the Application, and DEMO as the Function. Overwrite F to enter 0 as the minimum User Class that has access to this menu and bypass the next two fields (UID, password).

Enter MAILPROG as the first program name, followed by the description CUSTOMER MAILING LIST. Enter I as the program type because MAILPROG is an IDEAS program, accept 0 as the minimum User Class that can access the program, and bypass the next two fields.

Enter MAILREPT as the second program name, followed by the description MAILING LIST PRINTOUT. Enter I as the program type because this is also an IDEAS program, enter 0 as the minimum User Class that can access the program, and bypass the next two fields.

Bypass the remaining menu entry fields, since you have no other entries on this menu and no special entry you want to define as the CANCEL entry.

After you enter the above information, touch EXECUTE to accept the screen. The Menu Utility screen appears. You can touch FN 02 to document the menu if you wish. The menu documentation appears as follows.

Menu "MAILMENU" program documentation   09/27/56 Page 1 of 1

Revision number	0	Version	1	User class	0
Last revised	09/27/83	Application code	AAA	User ID code	
Last revised by	SJL	Function code	DEMO	Password	

Menu Screen Image

```
            1         2         3         4         5         6         7         8
  12345678901234567890123456789012345678901234567890123456789012345678901234567890
01 CUSTOMER MAILING LIST PROGRAM                                                  01
02                                                                                02
03                                                                                03
04                                                                                04
05                                                                                05
06                                                                                06
07                                                                                07
08                                                                                08
09                                                                                09
10                            FN Program Module                                   10
11                            00 CUSTOMER MAILING LIST                            11
12                            01 MAILING LIST PRINTOUT                            12
13                                                                                13
14                                                                                14
15                                                                                15
16                                                                                16
17                                                                                17
18                                                                                18
19                                                                                19
20                                                                                20
21                                                                                21
22                                                                                22
23                                                                                23
24 Touch any listed FN, SPACE/BACKSPACE to change default or EXEC to load default 24
  12345678901234567890123456789012345678901234567890123456789012345678901234567890
            1         2         3         4         5         6         7         8
```

Menu Selection Documentation

FN	Program	Description	Program type	UC	UID	Password
00	MAILPROG	CUSTOMER MAILING LIST	I	0		
01	MAILREPT	MAILING LIST PRINTOUT	I	0		
02	(unused)					
03	(unused)					
04	(unused)					
05	(unused)					
06	(unused)					
07	(unused)					
08	(unused)					
09	(unused)					
10	(unused)					
11	(unused)					
12	(unused)					
13	(unused)					
14	(unused)					
15	(unused)					
31	(unused)					

Note: UC = User Class
UID = User ID code

Program types: I = IDEAS2-generated program
W = Wang system software using R-range variables
X = Other software, requiring a complete COM CLEAR

56. Once the documentation is complete, the system returns to the Menu Utilities menu.

The final stage of this exercise is the writing of the report program. To enter the Report utilities, touch FN 11 from the Main Menu, then touch FN 00, and enter the name MAILREPT.

The Report Specification screen, Screen 35, appears.

```
IDEAS Report Generator - Main Specification Screen                  Release 2.1

Module name     MAILREPT                              Revision number
Description     ********************************      Last revised 09/27/83 SJL

Version               Edit & Document Privilege       Execution Privilege
Application           User class     User ID          User class     User ID
Function              Password                        Password

        Data files to be used in the module           Current record selection:
FN    File       Description                          Range   User modifiable
01                                                    Logic   User modifiable
02                                                    Sort    User modifiable
03                                                     by:
04
05
06
07

08  Save program REMs   N   Number of files    0      Number of fields          0
09  No. of lines/page  55   Work buffer     1750      Number of line formats    0
10  Program protection  N   Record buffer    256      Number of operations      0
```

Screen 35. Report Specification Screen.

57. Enter PRINTOUT OF MAILING LIST as the report description. Enter 1 as the Version, AAA as the Application and DEMO as the Function. Bypass the next six fields, accepting 0 as the user class when it appears. Enter MAILLIST as the Data file you want to use. Accept the screen.

When you first define a report, the next screen that appears is the Additional Data File Key Specification screen. This screen specifies the construction of random access keys for secondary associated data files. In this application, the report has only one associated data file, so you will not specify anything. Accept the screen as it is.

The Report Function Selection screen, Screen 36, appears.

```
IDEAS Report Generator - Summary & Function Selection Module          Release 2.1
Report program name is MAILREPT
Description PRINTOUT OF MAILING LIST

      File     Description                              ---- Summary ----
1  MAILLIST CUSTOMERS AND ADDRESSES              Number of fields              0
2                                                       alpha                  0
3                                                       numeric                0
4                                                Number of operations          0
5                                                Number of line formats        0
6                                                Maximum line length           0
7                                                Number of level breaks        0

FN Specification Options      Used  Modifiable   FN Specification Module    Used
'0 Field descriptions:         No   @ run time   --  --------------------   ----
   Record selection (Range)    No      No        '5 Line formats & fields    No
   Record selection (Logic)    No      No        '6 Operations               No
   Sort order specification    No      No        '7 File reading specs       Yes
'4 Default printer address     SYS     No        '8 Level break specs        No
(If default printer address is "SYS", address set in the START module is used)
Attention:      Touch FN'0-'8 listed above, EXECUTE to accept, FN'31 to cancel ■
```

Screen 36. Report Function Specification Screen.

58. The Report Function Selection screen is the screen from which you define all components of report creation and revision.

The box in the lower left quarter of the screen indicates record selection and printer selection options. You will not use the record selection options for this application because this report will only print in numeric order according to customer number. If you wish, you can touch FN 04, accept SYS, and change N(o) to Y(es) to allow the user to change the printer address at runtime. SYS as the printer address causes the address specified as the printer address in the START module for the application to appear as the default printer address at runtime.

The box in the lower right corner of the screen indicates four major options of report definition. You will not perform any operations on the fields of the report, nor will you use any special file reading specifications since this report only uses one key of one file. You will, however, use the other two options in this box, Line Formats and Fields, and Level Break Specs.

You define an IDEAS screen mask with a series of line formats. A line format may contain text or fields, and you can perform operations on the fields in a report in the same manner as on a screen. Each line format is identified as being of a particular level, and this level determines the format of the report.

To define line formats, touch FN 05. The Format Line Specification screen, Screen 37, appears.

```
IDEAS Report Generator - Report Mask Format Line Specifications    Release 2.1

Print control   Line Format Image Window         Cursor is at line  1, column  1
No T L FS B A   Window below currently displays lines  1 to 12, columns  1 to 65
01
02
03
04
05
06
07
08
09
10
11
12
                1↑                                                           65↑
T = Line type  | L = Level | FS = SYSFLAG# & state for | B = Extra line feeds
 B=Banner      |  Blank    |  optional sysflag test    |  before line is printed
 D=Detail      |  0-9      |  If used, line prints     | A = Extra line feeds
 F=Footer      |  P = Page |  only if sysflag state    |  after line is printed,
 H=Header      |  R = Report| AND level are correct.   |  in addition to the 1
 T=Test  N=Null|           |  State: "Y"=ON, "N"=OFF   |  issued automatically.
'0=Define/edit field, '16=Edit print control specs,  EXECUTE=accept
```

Screen 37. Format Line Specification Screen

59. Specify the following line formats in the following manner.

Move the cursor to Line 1, Column 1, and touch FN 16 to enter the print control specifications for this line.

a. Enter H as type. This indicates that the line is a Header.

b. Enter R as Level. This indicates the the line will print on a Report Level -- only at the beginning of the report, because it is a header; if it were a footer, it would print only at the end of the report.

c. Bypass FS (printing this line is not conditional based on a system flag), enter 0 for extra line feeds before printing, and enter 2 for extra line feeds after printing. Then enter the line format FAMILIARIZATION GUIDE SAMPLE REPORT in the Line Format Image Window.

Move the cursor to Line 2, Column 1, touch FN 16, enter H, P (page -- this is a page header and appears on every page), bypass FS, enter 0 for extra line feeds before printing and 2 for extra line feeds after printing. Enter the format MAILING LIST OF CUSTOMERS in the Line Format Image Window.

At column 44, enter PAGE, then move the cursor to Column 49 and touch FN 00 to define a field, which in this case is the system-defined field of current page number. Defining a field in a report is much like defining a field on a screen.

When you touch FN 00, the Report Field Specification screen, Screen 38, appears.

```
IDEAS Report Generator - Field Specifications                       Release 2.1

Report program name is: MAILREPT      #   File      Description
PRINTOUT OF MAILING LIST              1 MAILLIST  CUSTOMERS AND ADDRESSES
                                      2
Special (reserved) Field Names:       3
"@SYSPAGE" = Current page number      4
"@SYSMNTH" = System date month        5
"@SYSDAY " = System date day          6
"@SYSYEAR" = System date year         7

FN Attribute                  FN Attribute                FN Attribute
00 File number                06 Field type           2   11 Sign code
        name                     0 = unsigned numeric        0 = "-" before
01 Field name   FIELD001         1 = signed numeric          1 = "-" after
02 Line                2         2 = alphanumeric            2 = "DB" after (-)
03 Column             49      07 Justify              N      3 = "CR" after (-)
04 Length (memory)     0      08 Zero fill            N   12 Commas             N
          (report)     0      09 Decimals             0   13 Floating "$"       N
05 Position           80      10 Grouping level           14 Blank after print  N

Note: Special date field names yield 2-digit values.  Page number is 4 digits.
Attribute FNs shown above, EXEC=Accept, 16=Field names, 25=Delete, 31=Cancel
```

Screen 38. Report Field Specification Screen

60. Touch FN 01 and enter @SYSPAGE. This is a system-supplied default that provides the current page number of the report each time it is printed. Accept the screen and return to the Format Specification screen.

Line 3 is Type H, Level 1, with one line feed before and one line feed after. For a line format, enter CUSTOMER NUMBER. At Column 18, touch FN\00 to enter a field. Touch FN 00 again to enter 1 as the file number, enter CUST # as the field, and accept the screen.

Lines 4, 5, and 6 are all defined with the following characteristics in the following manner:

a. For all three lines, enter Type D after you touch FN 16. The system automatically bypasses the Level field for a Type D line. Bypass FS and bypass line feeds.

b. The format lines for 4, 5, and 6 consist only of fields from the data file. In Line 4, define a field at column 6, indicating 1 as the file number and NAME as the field name.

In Line 5, define a field at column 6, indicating 1 as the file number and ADDRESS as the field name.

In Line 6, define a field at column 6, indicating 1 as the file number and CITY as the field name. Similarly, define STATE at column 23. Define ZIP CODE at column 26; when defining Zip Code, however, touch FN 06 and change the field type to 2, alphanumeric. You must do this to ensure that leading zeros of the zip code will print. A report prints all numeric report fields as right-justified and space-filled, no matter how they are specified in the data file.

Move the cursor to Line 7 and touch FN 16 to define Line 7 as Type F, Level 1. Bypass FS, and specify one line feed before the line is printed and two line feeds after. At column 1, enter THIS CUSTOMER HAS A PRIORITY CODE OF. At column 38, touch FN 00 to define the field CLASCODE from File 1, MAILLIST.

Your screen should now look like the Complete Line Format screen, Screen 39.

```
IDEAS Report Generator - Report Mask Format Line Specifications    Release 2.1

Print control   Line Format Image Window          Cursor is at line  7, column  38
No T L FS B A   Window below currently displays lines  1 to 12, columns  1 to  65
01 H R    0 2   FAMILIARIZATION GUIDE SAMPLE REPORT
02 H P    0 2   MAILING LIST OF CUSTOMERS                          PAGE ####
03 H 1    1 1   CUSTOMER NUMBER  #####
04 D                 #########################
05 D                 #########################
06 D                 ################ ## #####
07 F 1    1 2   THIS CUSTOMER HAS A PRIORITY CODE OF #
08
09
10
11
12
               1↑                                                              65↑
T = Line type   L = Level   FS = SYSFLAG# & state for  B = Extra line feeds
 B=Banner        Blank        optional sysflag test       before line is printed
 D=Detail        0-9          If used, line prints     A = Extra line feeds
 F=Footer        P = Page     only if sysflag state       after line is printed,
 H=Header        R = Report   AND level are correct.      in addition to the 1
 T=Test   N=Null              State: "Y"=ON, "N"=OFF      issued automatically.
'0=Define/edit field, '16=Edit print control specs,  EXECUTE=accept
```

Screen 39. Completed Line Format Screen

61. Accept the screen and return to the Report Function Selection screen. Specify the level breaks for the printout to indicate where you want the headers and footers to print. Touch FN 08, then FN 01, then enter CUST #. This indicates that every time the field CUST # changes, the footers of Level 1 and below print in ascending order, then the headers of Level 1 and below print in descending order, then only detail lines print until the next level break occurs.

    Enter 1 as the file number, and accept No top-of-form. Accept the screen to return to the Report Function Specification screen (Screen 36). Accept the report. IDEAS then creates and saves a report control file. A message appears on Line 24 asking if you want to generate the program. Touch EXECUTE to generate the report program. After the program is generated, the system returns to the Report Utility screen.

62. To run the application, touch FN 15, enter MAILSTRT (if it is not already the default program), and then touch EXECUTE to begin execution. You can also run MAILSTRT directly by selecting the disk on which it resides and entering LOAD RUN "MAILSTRT".

    The program should load a menu with two options. First, run the Customer Mailing List option to put data in the data file, then run the Mailing List Printout option to print out the mailing list.

    The first time you choose the Mailing List Printout option, a screen appears asking you to create a sort work file. Accept the default disk address for this file and accept the default size of the default work file. If you specified in Step 58 that the printer address is modifiable at runtime, a screen appears allowing you to change the printer address. If you specify 005 as the printer address, the report appears on the CRT, one screen at a time, with all before and after line skipping specifications ignored.

INDEX

INDEX

WANG

Customer Comment Form

Publication Number 700-6872

Title 2200 IDEAS RELEASE 2 USER MANUAL

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 10-82-5C

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 1226A
ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851

Fold

Cut along dotted line.

WANG

To Order by Phone, Call:
(800)225-0234
From Mass., Hawaii, and Alaska
(617)256-1400
TELEX 951-743

Order Form for Wang Manuals and Documentation

① Customer Number (If Known)

② Bill To: Ship To:

③ Customer Contact: ④ Date Purchase Order Number

( ) ( )
Phone Name

⑤ Taxable Yes ☐ No ☐ ⑥ Tax Exempt Number ⑦ Credit This Order to A Wang Salesperson Please Complete Salesperson's Name Employee No. RDB No.

⑧ Document Number	Description	Quantity	⑨ Unit Price	Total Price

⑩ Authorized Signature Date

☐ Check this box if you would like a free copy of the **Corporate Publications Literature Catalog** (700-5294)

Sub Total	
Less Any Applicable Discount	
Sub Total	
LocalStateTax	
Total Amount	

Ordering Instructions

1. If you have purchased supplies from Wang before, and know your Customer Number, please write it here.
2. Provide appropriate Billing Address and Shipping Address.
3. Please provide a phone number and name, should it be necessary for WANG to contact you about your order.
4. Your purchase order number and date.
5. Show whether order is taxable or not.
6. If tax exempt, please provide your exemption number.
7. If you wish credit for this order to be given to a WANG salesperson, please complete.
8. Show part numbers, description and quantity for each product ordered.
9. *Pricing extensions and totaling can be completed at your option; Wang will refigure these prices and add freight on your invoice.*
10. Signature of authorized buyer and date.

Wang Supplies Division Terms and Conditions

1. **TAXES** — Prices are exclusive of all sales, use, and like taxes.
2. **DELIVERY** — Delivery will be F.O.B. Wang's plant. Customer will be billed for freight charges; and unless customer specifies otherwise, all shipments will go best way surface as determined by Wang. Wang shall not assume any liability in connection with the shipment nor shall the carrier be construed to be an agent of Wang. If the customer requests that Wang arrange for insurance the customer will be billed for the insurance charges.
3. **PAYMENT** — Terms are net 30 days from date of invoice. Unless otherwise stated by customer, partial shipments will generate partial invoices.
4. **PRICES** — The prices shown are subject to change without notice. Individual document prices may be found in the Corporate Publications Literature Catalog (700-5294)
5. **LIMITATION OF LIABILITY** — In no event shall Wang be liable for loss of data or for special, incidental or consequential damages in connection with or arising out of the use of or information contained in any manuals or documentation furnished hereunder.

Printed in U.S.A. 14-3141 7-83-5C

WANG

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 16 NO. CHELSMFORD, MA.

POSTAGE WILL BE PAID BY ADDRESSEE

WANG LABORATORIES, INC.
Supplies Division
c/o Order Entry Dept.
M/S 5511
51 Middlesex St.
No. Chelmsford, MA 01863

Fold

WANG

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

Printed in U.S.A.
700-6872
8-83-1.5M