



1st Edition, November 1980

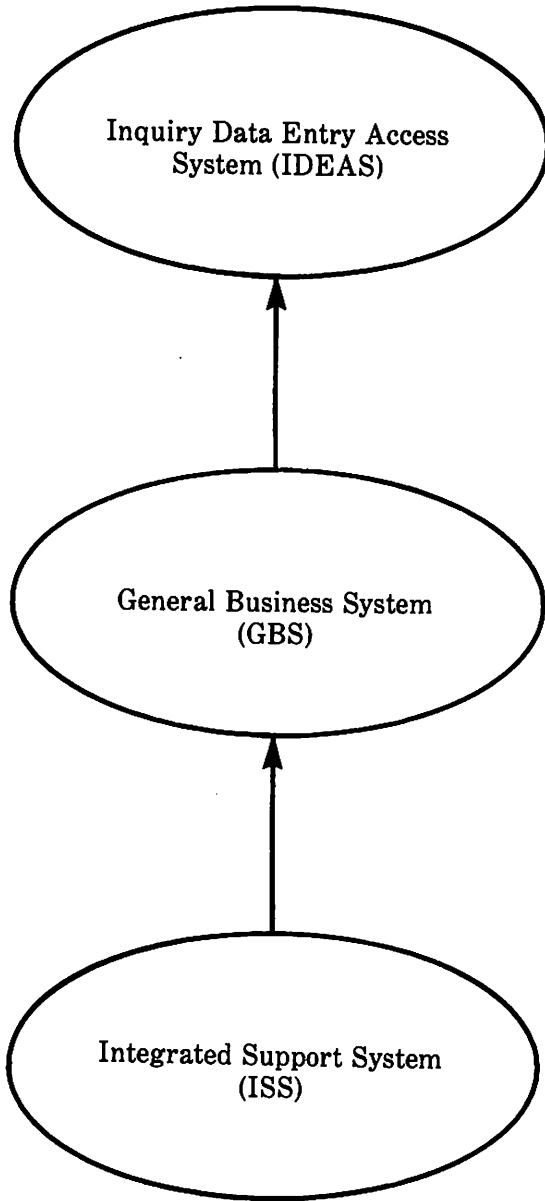
© 1980, Wang Laboratories, Inc.

For Internal Use Only

All Rights Reserved

2200 SOFTWARE PACKAGES

**Course Map**



CONTENTS

COURSE DESCRIPTION.....v  
    Prerequisites.....v  
    Course Objectives.....v

MATERIALS REQUIRED.....vi  
    Documentation.....vi  
    Facilities.....vi

COURSE ORGANIZATION AND DIRECTIONS FOR COMPLETION.....vii

MODULE 1: INTEGRATED SUPPORT SYSTEM (ISS).....TSS

MODULE 2: GENERAL BUSINESS SYSTEM (GBS).....TBS

MODULE 3: INQUIRY DATA ENTRY ACCESS SYSTEM (IDEAS).....TAS



## **COURSE DESCRIPTION**

The course, "Introduction to 2200 Software" (910-2002) presents information on three software packages that have been developed by Wang for use on 2200 Series computers. This course provides you with more detailed information on these three important software packages. They are Integrated Support System (ISS), General Business System (GBS), and Inquiry Data Entry Access System (IDEAS). These software packages have been instrumental in the success of Wang's 2200 product line.

ISS, GBS, and IDEAS each have a unique function and purpose. ISS is a system which provides programming and utility support. GBS is a series of application programs designed to meet the accounting and general business needs of most 2200 Series product line users. IDEAS is primarily a powerful program development tool which allows users to build sophisticated application programs very efficiently.

This course presents a description of the parts of each package, its specific features and examples of use. By gaining a fuller understanding of ISS, GBS, and IDEAS you will not only increase your own knowledge, but will also be able to better support these packages in the field.

### **Prerequisites**

Introduction to 2200 Hardware (910-2001)  
Introduction to 2200 Software (910-2002)

### **Course Objectives**

This course is intended to:

- Identify the major components of ISS, GBS, and IDEAS.
- Describe the purpose, features and benefits of each software package.
- Describe examples of the use of ISS, GBS, and IDEAS.
- Define hardware configurations and start-up procedures for ISS, GBS, and IDEAS.

## **MATERIALS REQUIRED**

### **Documentation**

ISS User Manual (700-5010A)  
Introducing GBS (700-4186D)  
IDEAS User Manual (700-5778)  
IDEAS Data Sheet (700-5747)

### **Facilities**

A quiet environment, free from frequent interruptions, for use as a study area.

## COURSE ORGANIZATION AND DIRECTIONS FOR COMPLETION

- This course consists of three individual modules. Each module represents a logical unit of content. The modules should be completed in the sequence indicated on the Course Map.
- Study in a quiet, comfortable place where you will not be distracted or interrupted.
- Set aside enough time to complete each module in a single study session.
- Gather all the materials and equipment you will need to complete a module before you begin a study session.
- Use the illustrations to reinforce the text. They can help you visualize and retain a mental picture of the material presented.
- Use the Exercises to test yourself as you progress. If you have trouble with one, stop to review the necessary module material. Once you have completed the Exercise, check your answers on the answer sheet provided. If you have missed any answers, feel free to go back and review the appropriate section.
- When you have completed the module text and the Exercises, contact the Course Administrator for the Module Test.



**MODULE 1**  
**INTEGRATED SUPPORT SYSTEM (ISS)**

CONTENTS

ABSTRACT.....TSS-1

OBJECTIVE.....TSS-1

MATERIALS REQUIRED.....TSS-2

DIRECTIONS FOR COMPLETION.....TSS-2

INTEGRATED SUPPORT SYSTEM (ISS).....TSS-3  
     Benefits of ISS.....TSS-4

ISS-5 START-UP PROCEDURES.....TSS-5

ISS UTILITIES.....TSS-11  
     Copy/Verify Utility.....TSS-12  
     Create Reference File Utility.....TSS-15  
     List/Cross Reference Utility.....TSS-17  
     Compression Utility.....TSS-20  
     Decompression Utility.....TSS-22  
     Sort Disk Catalog Utility.....TSS-24  
     Disk Dump Utility.....TSS-27  
     File Status Report Utility.....TSS-29  
     Program Compare Utility.....TSS-30  
     Reconstruct Disk Index Utility.....TSS-32  
     Alter Disk Index Utility (ISS-5 Release Only).....TSS-33  
     Summary.....TSS-34

REVIEW QUESTIONS.....TSS-35

ANSWERS.....TSS-36

SCREEN/DISK SUBROUTINES.....TSS-37  
     Data Entry Subroutine (DEFFN'200).....TSS-39  
     Data Routines (DEFFN'220, 221, 223, 224, 225).....TSS-39  
     Operator Wait Subroutine (DEFFN'254).....TSS-41

DISK ROUTINES.....TSS-42  
     Search Index Subroutine (DEFFN'229).....TSS-43  
     Allocate Data File Space Subroutine (DEFFN'228).....TSS-43  
     Free Unused Sectors Subroutine (DEFFN'227).....TSS-43  
     Limits Next Subroutine (DEFFN'226).....TSS-44  
     Open/Close Input/Output Subroutines (DEFFN' 240, 241,  
         250, 251).....TSS-45  
     Multiplexed/Multistation File Open/End/Close  
         Subroutines (DEFFN' 217, 218, 219).....TSS-45  
     Translation Tables.....TSS-47  
     Summary.....TSS-48

CONTENTS (CONT)

SORT-4.....TSS-49  
    Summary of Sort-4 Features.....TSS-49  
    Sort Comparisons.....TSS-51  
    Summary.....TSS-53

KFAM (KEY FILE ACCESS METHOD).....TSS-55  
    Componenets of KFAM.....TSS-57  
    General Purpose Subroutines.....TSS-57  
    Random Access Subroutines.....TSS-58  
    Key Sequence Access Subroutines.....TSS-58  
    Add and Delete Subroutines.....TSS-58  
    Special Purpose Subroutines.....TSS-59

SUMMARY.....TSS-60

REVIEW QUESTIONS.....TSS-62

ANSWERS.....TSS-64

## ABSTRACT

The format of this module is different from previous ones you have completed. The difference exists in the presentation of the various parts of the ISS utilities and screen/disk subroutines. There are, for example, 11 ISS utilities and 10 screen/disk subroutines; a narrative treatment of each of these would be lengthy and taxing on the reader. To simplify this presentation, a standard format is used which effectively presents information on each program. The format consists of four parts: purpose, use, procedure and example (a visual example, if possible). This format has the advantage of being a handy reference guide.

The Integrated Support System (ISS) is a highly versatile system developed for the 2200 Series product line. As its name suggests, ISS provides programming and utility support through its file access utility functions, and pre-defined subroutines. The purpose of this module is to familiarize you with the specific features of ISS so that you can assist customers in utilizing the benefits of the ISS package.

## OBJECTIVES

At the completion of this module, you will be able to:

- Discuss the general features and benefits of ISS.
- Know the step-by-step procedures for ISS start-up.
- Describe each of the ISS utilities in terms of purpose, use, procedure, and example.
- Describe each of the screen/disk subroutines in terms of purpose, use, procedure, and example.
- Describe SORT-4 in terms of purpose, use, procedure and example.
- Describe the evolution and purpose of KFAM.
- Describe the key subroutines and utilities of KFAM in terms of purpose, use, procedure, and example.

## INTEGRATED SUPPORT SYSTEM

---

### MATERIALS REQUIRED

ISS User Manual (700-5010)

The key reference document for this module is the ISS User Manual. It is used extensively in the field, and customers often refer to it when they have specific questions. This manual is your primary resource for detailed answers to technical questions on ISS.

### DIRECTIONS FOR COMPLETION

After completing this Module, contact the Course Administrator for the Module Test.

INTEGRATED SUPPORT SYSTEM (ISS)

ISS provides a wide range of programming support and fills a number of data processing needs for 2200 Series computer customers. ISS accomplishes these tasks through its file access software, utility functions, and pre-defined screen/disk subroutines. File access is handled by a Key File Access Method (KFAM) which offers rapid access to data by means of subroutines capable of handling both random and sequential access. The utility programs are user-controlled routines which allow program files to be copied, compressed, decompressed, listed, sorted, cross-referenced, and compared to other files. Special purpose utility functions allow creating, editing or printing a reference file\*, as well as displaying or printing the contents of a data file. Screen/disk subroutines perform standard programming tasks related to either user/screen or program/disk interaction; these subroutines greatly reduce an application programmer's need to write repetitious, detailed routines.

There are four main components of ISS. Figure 1 displays these components in order of their treatment in this module and in the ISS User Manual.

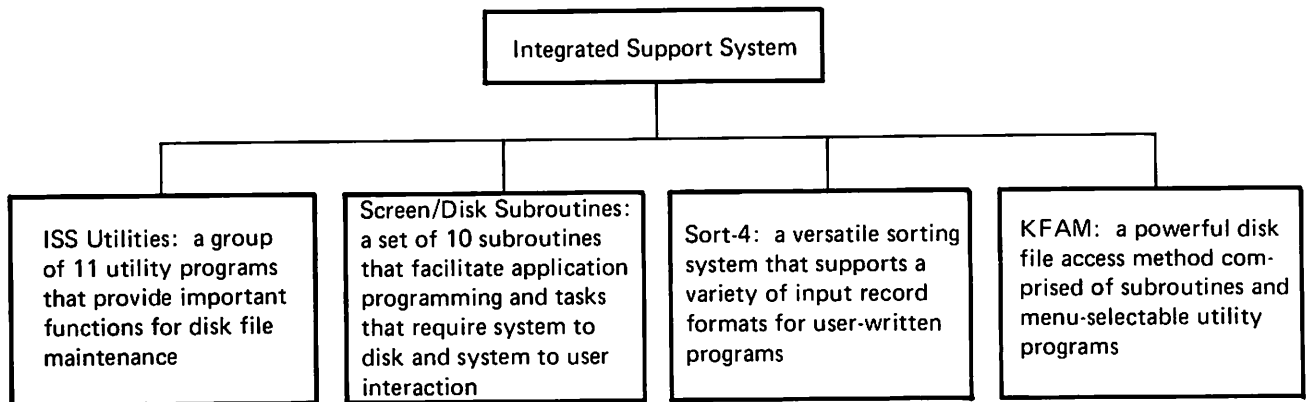


Figure 1. The Major Components of ISS

\* A reference file is a file that contains multiple entries in the form of an input file name, an output file name, and an extra sector's value. Reference files are used by a number of ISS utilities (e.g., copy/verify is a means of specifying a frequently used set of file names).

# INTEGRATED SUPPORT SYSTEM

---

## Benefits of ISS

There are several important benefits of the ISS package to the customer. First, it offers a comprehensive support system. Many frequently used utilities and subroutines have been combined into a single comprehensive system. Second, ISS is economical because packaged programs save development time and money. Additional time is saved because ISS is largely menu-driven and easy to use for programmers. And third, ISS can increase productivity because it allows programmers to focus on application and data processing needs, rather than spending time developing the types of common, frequently used routines that are provided in ISS. It is important for two reasons, to recognize that ISS is now in its fifth release. One is that it illustrates that Wang continues to update and improve products to better serve its customers' data processing needs. ISS is a tried and proven system, and Wang is continually searching for ways to make ISS more comprehensive, economical, and productive for the customer. The second reason is that there are CPU compatibility exceptions for different ISS releases. The following table presents comparative data for ISS releases 3 and 5. When specific questions on ISS releases arise, the answers can be found in the ISS User Manual.

Table 1. Software and CPU Support for ISS Releases 3 and 5

Description	ISS-3	ISS-5
<u>Software</u>		
Start-up	.	.
Program Support Utilities*	.	.
Screen/Disk Subroutines	.	.
Sort-4	.	.
KFAM-5	.	.
KFAM-7	.	.
<u>CPU Support</u>		
2200T	.	.
PCS-II	.	.
PCS-III	.	.
2200SVP	.	.
2200VP	.	.
2200LVP	.	.
2200MVP	.	.

\*The Utility programs included in ISS-5 are generally more flexible and have more features than those in ISS-3. Also, 1 new utility has been added.



ISS-5 START-UP PROCEDURES

Start-up procedures must be completed before any of the ISS utilities or subroutines can be used. Each station begins ISS start-up via a program file called START. During start-up operations, the user defines available peripherals and the current date; this information can be accessed by any ISS software running in that "station." For single-task computers, the central processor is considered to be a station. For multi-task computers, each memory partition is a station. After start-up operations, the designated ISS menu is displayed and ISS is fully operational.

ISS start-up typically is necessary at the start of a day or when first running ISS after another application.

Step 1: MVP PARTITION GENERATION\*

In order to load and start ISS, there are several actions that must be taken in normal MVP partition generation. The first action is to ensure that the partitions are large enough for the ISS components desired. Several examples of partition size are presented in Table 2. A complete listing of the amount of memory needed for ISS utilities and subroutines is in the ISS User Manual.

Table 2. Partition Size for Selected ISS Components

<u>ISS Component</u>	<u>Partition Size Requirements</u>
ISS Start-up	8.5
Copy/Verify	10
List/Cross Reference	12
Compression	13.5
Disk Dump	9
Screen/Disk Subroutines	12
SORT-4	9 or 12
KFAM-7 Utilities**	9K each

\* For the purpose of this module, the MVP is used as the example for ISS start-up. These procedures also apply to the LVP CPU.

\*\* 1.4K is required within each partition using KFAM for the local KFAM variables.

A second partition generation consideration is the Master Device Table. Any peripheral device that will be used during ISS operation must have its address entered in the Master Device Table. If that address has not been entered, choose the "Edit Device Table" option on the partition generation menu.

Step 2: LOADING ISS START-UP SOFTWARE

Before mounting the ISS diskette (For T, VP, or MVP users, this will be one of four ISS diskettes, depending on the application desired.), be sure the tab is in place over the Write/Protect notch. Mount the diskette and enter the following sequence.

```
SELECT DISK XYZ
LOAD RUN
```

"XYZ" is the disk device address where the ISS diskette has been mounted. LOAD RUN automatically loads the START program into memory and runs it. This initiates ISS.

Step 3: RESPONSE TO FIRST PROMPT IN ISS START-UP

The first prompt is "ENTER STATION NUMBER" and there are three response options:

- Option 1 - SF Key 0 to view existing and/or default station file numbers.
- Option 2 - SF Key 16 to create a station file for a particular station. A prompt then appears "ENTER STATION NUMBER TO CREATE"; values from 1 to 48 are legal entries.
- Option 3 - Enter a station number from 1 to 48 to proceed to the next series of prompts on system configuration. If the number entered is unacceptable, this indicates that the station file for this station does not exist (in this case, refer to Option 2 above).

Step 4: RESPONSE TO PROMPTS ON DEFAULT/NO DEFAULT VALUES IN STATION FILE

The next series of prompts is displayed when Option 3 is properly executed. When default values are in the station file, one series of prompts is presented. When no default values are in the station file, another series appears. Each of these cases is treated in turn.

- No Default Values In Displayed Station File.

The following screen appears when the station file is blanked. This situation usually occurs when a new station is being created.

```
ENTER TODAYS DATE (MM/DD/YY)
-
STATION # 3
1 DATE -
2 MENU TO LOAD -
3 PRINTER ADDRESS -
4 LOADING ADDRESS -
5 DISK ADDRESSES -
```

Figure 2. Screen Resulting from Blanked Station File

After typing in the data (in response to "Enter Today's Date" prompt) and pressing EXECUTE, the "Printer Address" prompt appears. This prompt is accompanied by a set of allowable printer addresses. You can then specify that you want to use a local printer (address 204), a system printer (address 215, for example), or no printer at all (printout will be displayed on CRT only).

After entering the printer address, the "Loading Address" prompt appears. Enter the disk address where ISS resides.

Next is the "Disk Addresses" prompt, accompanied by a set of allowable addresses. Enter the device addresses you will need for functions to be carried out by that station.

Finally, the "Menu to Load" prompt appears. Enter one of the six options (including Utilities, Screen/Disk subroutines, etc.) which appear on the screen.

- Default values in the Displayed Station File. When default values have been loaded from the station or manually entered from the keyboard, they are presented in the following format:

```
ENTER DESIRED FUNCTION (0 = END)
-
STATION # 3
1 DATE - 08/20/80
2 MENU TO LOAD - UTILITIES
3 PRINTER ADDRESS - 215
4 LOADING ADDRESS - 310
5 DISK ADDRESSES - B10 320 B20
```

Figure 3. ISS Start-Up Screen - With Default Values

In this situation, some or all of the default values may need to be changed. The prompt at the top of the screen asks the user to specify which values should be changed. For example, when only a disk address needs to be modified, 3 is the proper response to the "Enter Desired Function" prompt. When all the information is correct, 0 is pressed and the menu selected in the "Menu to Load" choice is displayed.

When Start-up has been successfully completed, the "Menu to Load" is displayed. When the ISS System Menu is chosen it resembles Figure 4. The System Menu can be accessed easily during ISS operation; it allows quick, direct entry into the major ISS components. The following discussion of ISS includes each of the menu options with the exception of "Applications." The applications menu allows a user-created application program to be loaded automatically and run.

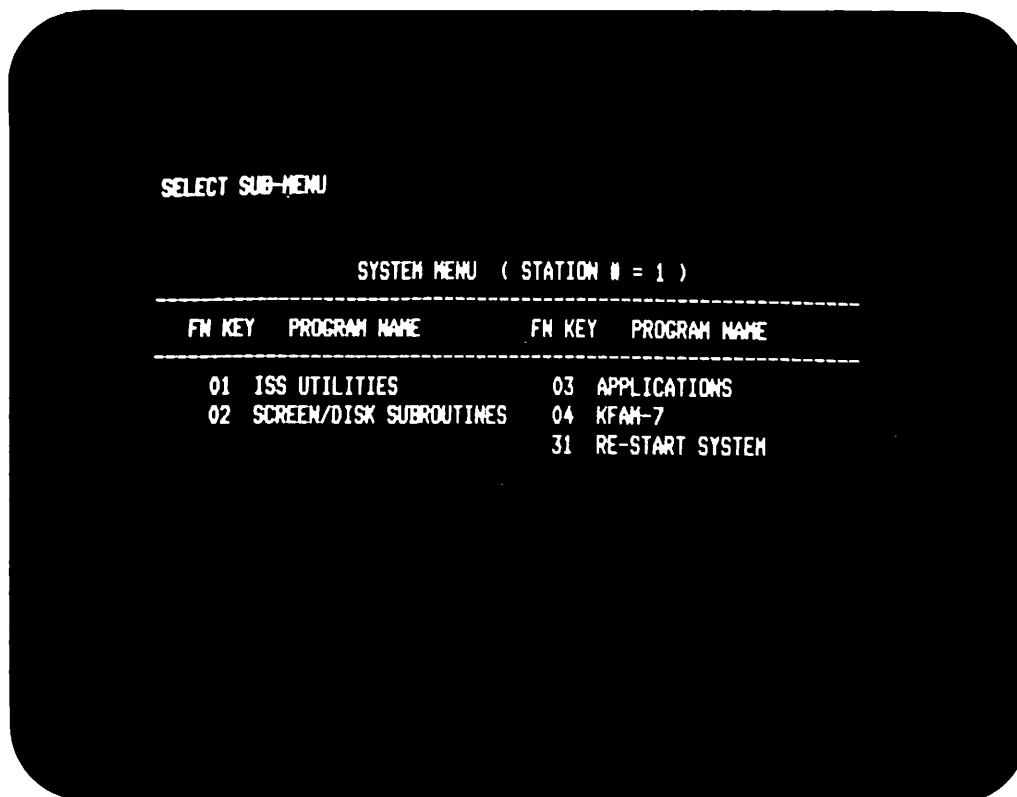
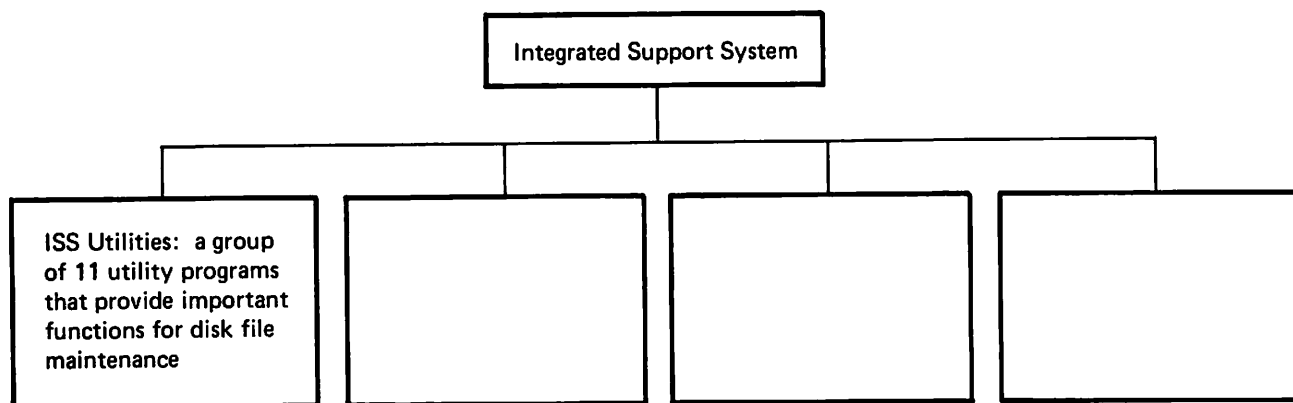


Figure 4. ISS System Menu

ISS UTILITIES



The ISS UTILITIES are an important part of the total ISS software package. The eleven utilities are operator-controlled programs which provide various programming support and standard disk-related functions. The ISS Utilities are widely used and very popular with customers. As with all ISS Components, the Utilities have been continually improved and modified to provide users with the best possible utility support. In ISS Release 5, a new program, Alter Disk Index, has been added to the ISS utilities.

```

SELECT UTILITY

ISS UTILITIES ( STATION # 1 )
-----
FN KEY   PROGRAM NAME           FN KEY   PROGRAM NAME
-----
00 COPY/VERIFY           05 SORT DISK CATALOG
01 CREATE REFERENCE FILE 06 DISK DUMP
02 LIST/CROSS-REFERENCE 07 FILE STATUS REPORT
03 COMPRESSION           08 PROGRAM COMPARE
04 DECOMPRESSION         09 RECONSTRUCT DISK INDEX
                          10 ALTER DISK INDEX
                          31 SYSTEM MENU
    
```

Figure 5. ISS Utilities

## INTEGRATED SUPPORT SYSTEM

These eleven programs can be grouped into functional categories which highlight commonalities among the various utility programs. The four categories presented in Table 3 are Copy, Programming, Catalog Index, and Special Purpose Functions.

Table 3. Functional Categories of ISS Utilities

Functional Category	Storage Element	Utility
Copy Function	All Files	Copy/Verify
Programming Functions	Program Files Program Files Program Files Program Files	List/Cross Reference Compression Decompression Program Compare
Catalog Index Functions	Disk Disk Disk	Sort Disk Catalog Reconstruct Disk Index Alter Disk Index
Functions Special Purpose	All Files Data Files All Files	Disk Dump File Status Report Create Reference File

A detailed treatment of the Utilities is contained in the ISS User Manual. In these training materials, each utility is discussed in terms of purpose, use, key procedures, and an example.

### Copy/Verify Utility

**Purpose:** Copies standard 2200 cataloged files from one disk to another with the option of allocating extra space to a file. Copied files may replace existing files or be stored as new files. The copied output files may be optionally verified.

**Use:** This utility has a number of common uses. One of the most frequent is to make a backup copy of important programs or data files. (Copy/Verify, for example, was used to make a copy of the system diskette in the lab portion of the Introduction to 2200 Software course.)



## Procedures:

- The specified file can be copied, verified, or copied and verified.
- In general, the content of each input file -- up to and including the END trailer -- is copied into the output file. The user may specify that extra sectors beyond the END trailer be included.
- Output Options.
  - ADD - This will copy the file to the output disk only if it currently does not exist on that disk.
  - REPLACE - This will copy the file to the output disk only if it currently exists on that disk, in effect replacing the file.
  - ADD/REPLACE - This copies the file to the output disk unconditionally.
- MODE determines which active files on the input disk are to be processed. The options include ALL, PART, RANGE, INDIRECT, PROGRAM, and DATA; these options are defined in the ISS User Manual.

# INTEGRATED SUPPORT SYSTEM

Example: In this case, (exhibited in Figure 6) part of the files on B10 are to be copied and verified on to disk address 320.

```
ENTER DESIRED FUNCTION      (0 = END)
-
                                COPY/VERIFY ( STATION # 1 )
1 INPUT ADDRESS - B10          4 OUTPUT ADDRESS - 320
2 INPUT OPTION - COPY/VERIFY   5 OUTPUT OPTION - ADD/REPLACE
3 MODE - PART                  6 EXTRA SECTORS - -1
7 SAVE DEFAULTS                '15 RETURN TO CHANGE PARAMETERS
```

Figure 6. Copy/Verify Screen

## Create Reference File Utility

**Purpose:** Creates, edits (from the contents of a disk catalog), or prints a system reference file. A system reference file specifies input file name, output file name, and number of extra sectors.

**Use:** The reference file created by this utility is often used in conjunction with Copy/Verify when portions of one disk are to be copied onto a different disk. Once the reference file is created, it can be used by any utility which supports the INDIRECT mode (i.e., a reference file is on the same disk as the input files). Other utilities which use Reference Files are Decompression, Compression, and Program Compare.

### Procedures:

- Create, edit, or print options can be chosen.
- When Create or Edit are selected, there are two distinct operating phases: 1) selecting the input file names to be copied, and 2) specifying the output file names and extra sector values. (Default names for output files are the same names as input files.)
- A "window" of ten active files is displayed on the CRT. This window is manipulated by SF Keys and its purpose is to select those files which will be incorporated into the reference file.

**Example:** In Figure 7, the input file names are being selected from the disk catalog index. Notice that EASYSCR1 was selected and \$KALAIDS was not. The user is about to make a choice on IDFU-3MO. In Figure 8, several of the output and extra sector values have been specified. Notice that input file name "ADDRESS" has been changed to "LOCATION."

```

FN'0 - INDEX CONTROL      FN'16 - END      DISK ADDRESS = 320
FN'1 - FILE CONTROL      FILE NAME = REF8

INDEX                     INPUT      OUTPUT      X-SEC
-----
GPJII P                   0006 DEMSCREN
INVT090A P                0007 IOFU-305
NETCALL P                 0008 CREP0002
#KALAIOS P                0009 NETCALL
EASYSCR1 P                0010 EASYSCR1
IOFU-3M0 P (<=====>) 0011
MONITOR1 P                0012
GAMES P                   0013
CONVERT P                 0014
KFAH1007 P                0015

```

Figure 7. Reference File Utility Screen

```

FN'0 - INDEX CONTROL      FN'16 - END      DISK ADDRESS = 320
FN'1 - FILE CONTROL      FILE NAME = REF9

INDEX                     INPUT      OUTPUT      X-SEC
-----
INVT0600 P                0002 INVT100A INVT100A  006
INVFILE P                 0003 GAP20   GAP20     001
INVT110B P                0004 AUDIFILE AUDIFILE  001
CONVERT P                 0005 ACCTMIKE ACCTMIKE  015
UTIL020A P                0006 DEMORPT  DEMOREPT  005
INVT300A P (<=====>) 0007 ADDRESS  *LOCATION
GEMLF010 D                0008 ACPAMENU
SLOTS P                   0009 MORTGAGE
IDPROG06 P                0010 UTIL020A
IDFU-301 P                0011

```

Figure 8. Reference File Utility Screen

## List/Cross Reference Utility

**Purpose:** Lists and/or creates a cross reference for a specified program file. The List component prints the contents of each program file. The Cross Reference component builds three cross reference tables:

- A list of each referenced line number and each other line number which references it.
- A list of line numbers which contain each variable.
- A list of each DEFFN statement's line number along with lines which reference (via a GOSUB statement) that DEFFN statement.

**Use:** This is an extremely valuable utility for the programmer, especially debugging. It is also a useful method for documentation.

### Procedures:

- The three options for this utility are List, Cross-reference, and List/Cross-reference.
- Expanded titles can be ordered with the List option.
- Any protected program file is not eligible for this utility.

**Example:** In Figure 9, a program named ARITH is listed. This is a program that was included on the diskette in the "Introduction to Software" Course. Figure 10 displays four cross-reference tables. The first table shows line numbers referenced within the program. This shows that line 30 is accessed at six different lines within the program (twice in the line 80 statements and four times in line 90 statements). The next table lists line numbers where each numeric or alphanumeric variable is referenced. The line numbers where marked subroutines are used are then listed; and finally, a summary chart is displayed.

```

4 DEFFN'14"PRINTHEX(03) :SELECT LINE 22 :LIST S D"
9 DEFFN'31"SCRATCH F/310,";HEX(22);"ARITH";HEX(22);":SAVE DC <S>F/310,()" ;HEX(22);"ARITH
";HEX(22)
10 REM Program name ARITH 5 Jun 1980 (JKK for KCS) Called by Menu Developed for Spectr
um Training Corp, Boston, MA
20 %The numbers are -###,###,###,###.##### and -###,###,###,###.#####
30 %##### -###,###,###,###,###.#####
40 PRINT HEX(03);AT(1,20);"It's nice to work with you, ";N$
:PRINT AT(3,8);"This program will give you a little practice using the keyboard"
:PRINT AT(5,6);"To return to the menu at any time press the FN key on the left side"
50 PRINT AT(8,15);"Let's try a a very simple number operation"
:PRINT AT(10,5);"It doesn't matter whether you use the 10-key pad or the numbers"
:PRINT AT(12,15);"on the typewriter keyboard for the input"
60 A=0
:PRINT AT(14,20);
:INPUT "Any number ",A
70 B=0
:PRINT AT(16,22);"Any positive number";AT(16,45,15);
:INPUT B
:IF B>0 THEN 80
:PRINT HEX(07);AT(19,27);"DIDN'T THINK I'D CATCH YOU DID YOU ?"
:GOTO 70
80 PRINT HEX(03)
:PRINTUSING 20,A,B
:PRINT AT(3,1);
:PRINTUSING 30,"Sum is",A+B
:PRINT AT(5,1);
:PRINTUSING 30,"Difference is",A-B
90 PRINT AT(7,1);
:PRINTUSING 30,"Product is",A*B
:PRINT AT(9,1);
:PRINTUSING 30,"Quotient of the 1st divided by the 2nd is",A/B
:PRINT AT(11,1);
:PRINTUSING 30,"Square of the 1st is",A*A
:PRINT AT(13,1);
:PRINTUSING 30,"Square root of the 2nd",SQR(B)
100 PRINT AT(16,3);"If the results show some # signs, the numbers were too large to print"
:PRINT AT(18,15);"Don't you wish you were as fast as I am ????"
:PRINT AT(20,7);"Press RETURN to do it again or use FN key to branch to menu ";
:INPUT X$
:PRINT HEX(03)
:GOTO 40
110 DEFFN'126
:SELECT P5
:PRINT HEX(03);AT(10,22);"Loading Menu for ";N$
:SELECT P
:LOAD DC T#1,"Menu" 10,
900 PRINT HEX(03);AT(12,12.12);
:INPUT X
910 PRINT AT(12,12.12);
:INPUT X
:GOTO 910

```

Figure 9. Listing of "ARITH"

07/22/80  
LINE NUMBER CROSS-REFERENCE

ARITH

PAGE 1

10 - < 1> -- 110  
 20 - < 1> -- 80  
 30 - < 6> -- 80 80 90 90 90 90  
 40 - < 1> -- 100  
 70 - < 1> -- 70  
 80 - < 1> -- 70  
 910 - < 1> -- 910

07/22/80  
VARIABLE CROSS-REFERENCE

ARITH

PAGE 2

A - < 9> -- 60 60 80 80 80 90 90 90 90  
 B - < 9> -- 70 70 70 80 80 80 90 90 90  
 X - < 2> -- 900 910

\*\*\*\*\*

N\$ - < 2> -- 40 110  
 X\$ - < 1> -- 100

07/22/80  
MARKED SUBROUTINE CROSS-REFERENCE

ARITH

PAGE 3

DEFFN' 14< 1> -- 4  
 DEFFN' 31< 1> -- 9  
 DEFFN' 126< 1> -- 110

07/22/80  
SUMMARY

ARITH

PAGE 4

TEXT LINES = 15                      TEXT STATEMENTS = 50  
 LINE NUMBERS = 7                    VARIABLES = 5  
 MARKED SUBROUTINES = 3

Figure 10. Cross-Reference Output for Program "ARITH"



### Compression Utility

**Purpose:** Reduces the amount of memory required by a program and increases execution speed by eliminating spaces, unessential line numbers, and REM (remark) statement lines.

**Use:** This utility can be used for any program; it saves storage space, execution time, and otherwise encourages the economical use of the computer.

#### Procedures:

- On a file-by-file basis, each specified input program file is read, compressed, and copied to the specified output file.
- All REM statements are eliminated, except those in the first line of the program.
- All space characters (blanks) are eliminated except those within quotation marks or an image % statement.
- Unnecessary line numbers are eliminated by appending as many BASIC-2 statements as possible onto a single line. Certain BASIC statements and statement lines are not appended. (See ISS User Manual.)

**Example:** The space that this utility saves is obvious when Figures 11 and 12 are compared. Notice that the first REM statement remains but all others are eliminated.

```

READY (BASIC-2) PARTITION 04
:LIST
10 REM   *** DEMONSTRATING THE COMPRESSION UTILITY ***
20 PRINT
30 REM   A SIMPLISTIC PROGRAM FOR CREATING A NEW FILE AND
40 REM   SAVING RECORDS INTO IT
50 DIM N$10,D$40,S$6
60 REM   *** ESTABLISH A FILE AND OPEN IT ***
70 DATA SAVE DC OPEN F 200, "INVTORY"
80 REM   *** ENTER DATA FROM ONE INVENTORY RECORD ***
90 INPUT "PRODUCT NUMBER",N$
100 INPUT "PRODUCT DESCRIPTION",D$
110 INPUT "SUPPLIER CODE",S$
120 INPUT "ON-HAND QUANTITY",Q
130 INPUT "INDICATED REORDER LEVEL",R
140 REM   *** SAVE RECORD ON DISK ***
150 DATA SAVE DC N$,D$,S$,Q,R
160 REM   *** MORE RECORDS? ***
170 INPUT "MORE PRODUCTS (Y OR N)",R$
180 IF R$="Y" THEN 90

:

```

Figure 11. Initial Program

```

READY (BASIC-2) PARTITION 04
:LIST
10 REM   *** DEMONSTRATING THE COMPRESSION UTILITY ***
20 PRINT : DIM N$10,D$40,S$6: DATA SAVE DC OPEN F200,"INVTORY"
90 INPUT "PRODUCT NUMBER",N$: INPUT "PRODUCT DESCRIPTION",D$: INPUT "SUPPLIER CO
DE",S$: INPUT "ON-HAND QUANTITY",Q: INPUT "INDICATED REORDER LEVEL",R: DATA SAVE
DC N$,D$,S$,Q,R: INPUT "MORE PRODUCTS (Y OR N)",R$: IF R$="Y"THEN 90

:

```

Figure 12. Compressed Program

**Decompression Utility**

- Purpose:** Separates compressed or multistatement program lines into single statement lines.
- Use:** Compressed program listings are difficult to read and work with. Therefore, it is frequently necessary to expand and space out listings of programs that have used the compression utility. The utility is also used on programs that are initially tightly written by the programmer.
- Procedures:** Some multistatement lines may remain if certain BASIC 2 statements are used or if the supply of available line numbers is exhausted.
- Example:** In the decompressed version, the program is back to its single statement/line format, but the line numbers are different from the initial program (See Figure 11). Also, of course, the original REM statements cannot be restored.

```

READY (BASIC-2) PARTITION 04
:LIST
10 REM   *** DEMONSTRATING THE COMPRESSION UTILITY ***
20 PRINT : DIM N$10,D$40,S$6: DATA SAVE DC OPEN F200,"INVTRY"
90 INPUT "PRODUCT NUMBER",N$: INPUT "PRODUCT DESCRIPTION",D$: INPUT "SUPPLIER CO
DE",S$: INPUT "ON-HAND QUANTITY",Q: INPUT "INDICATED REORDER LEVEL",R: DATA SAVE
DC N$,D$,S$,Q,R: INPUT "MORE PRODUCTS (Y OR N)",R$: IF R$="Y"THEN 90

:

```

Figure 13. Compressed Listing

```

READY (BASIC-2) PARTITION 04
:LIST
10 REM   *** DEMONSTRATING THE COMPRESSION UTILITY ***
20 PRINT
21 DIM N$10,D$40,S$6
22 DATA SAVE DC OPEN F200,"INVTRY"
90 INPUT "PRODUCT NUMBER",N$
91 INPUT "PRODUCT DESCRIPTION",D$
92 INPUT "SUPPLIER CODE",S$
93 INPUT "ON-HAND QUANTITY",Q
94 INPUT "INDICATED REORDER LEVEL",R
95 DATA SAVE DC N$,D$,S$,Q,R
96 INPUT "MORE PRODUCTS (Y OR N)",R$
97 IF R$="Y"THEN 90

:

```

Figure 14. Decompressed Listing

### Sort Disk Catalog Utility

**Purpose:** Prints or displays all entries on the disk catalog index in alphabetical or disk sector sequence. The information that is printed includes each file's name, beginning and ending sector address, and the number of used and free sectors.

**Use:** After a great deal of use, the disk catalog can become unwieldy to reference. This utility allows the catalog to be coherently sequenced and therefore much easier to use.

#### Procedures:

- Active files, scratched files, or both (all files) may be specified for the file list.
- Output can be either on the CRT or on a designated printer.

**Example:** Examples are shown of the catalog (see Figure 15) being sorted alphabetically by file name (Figure 16) and sequentially by starting sector address (Figure 17). In Figure 16, notice that the file names beginning with symbols are placed first and are sorted alphabetically on the next character. If RETURN is keyed, Figure 18 (which summarizes the status of the catalog index) is displayed.

```

HEAD: BASIC-D
LIST OCR

INDEX SECTORS = 00011
END CAT. AREA = 01221
CURRENT END = 00277

NAME      TYPE  START  END  USED  FREE
MORTGAGE  P    00213 00220 00006 00002
UTIL      P    00239 00247 00004 00005
MATSORT1  P    00090 00106 00008 00005
PAYCNTB   P    00204 00212 00004 00005
IDISPLAY  P    00248 00277 00022 00008
@SYSFILE  D    00011 00035 00024 00001
INVT220A  F    00036 00085 00049 00005
@PSTAT    F    00221 00238 00013 00005
ACCTHENU  P    00107 00136 00025 00005
SALEFILE  G    00137 00203 00067 00000
    
```

Figure 15. Unsorted Catalog

```

INDEX SORTED BY NAME
KEY RETURN(EXEC) TO RESUME?

DISK - B10          DATE - 07/22/80          PAGE 1

INDEX SECTORS = 11
END CAT. AREA = 1221
CURRENT END = 277

ITEM  NAME      TYPE  START  END  USED  FREE
  1  IDISPLAY  P    248  277  22  8
  2  @PSTAT    P    221  238  13  5
  3  @SYSFILE  D     11  35  24  1
  4  ACCTHENU  P    107  136  25  5
  5  INVT220A  P     36  89  49  5
  6  MATSORT1  P     90  106  8  9
  7  MORTGAGE  P    213  220  6  2
  8  PAYCNTB   P    204  212  4  5
  9  SALEFILE  D    137  203  67  0
 10  UTIL      P    239  247  4  5
    
```

Figure 16. Alphabetically Sorted Disk Catalog

```

SORTED BY STARTING SECTOR
KEY RETURN(EXEC) TO RESUME? _
DISK - B10          DATE - 07/22/80          PAGE 1

INDEX SECTORS = 11
END CAT. AREA = 1221
CURRENT END   = 277

ITEM  NAME      TYPE  START  END   USED  FREE
  1   @SYSFILE  D     11    35   24    1
  2   INVT220A  P     36    89   49    5
  3   MATSORT1  P     90   106    8    9
  4   ACCTMENU  P    107   136   25    5
  5   SALEFILE  D    137   203   67    0
  6   PAYCNTB   P    204   212    4    5
  7   MORTGAGE  P    213   220    6    2
  8   @PSTAT    P    221   238   13    5
  9   UTIL      P    239   247    4    5
 10   $DISPLAY  P    248   277   22    8

```

Figure 17. Sorted by Starting Sector

```

INDEX SORTED BY NAME
KEY RETURN(EXEC) TO RESUME? _
SUMMARY OF DISK USAGE

          PROGRAM FILES  DATA FILES
ACTIVE          8          2
SCRATCHED       0          0
TOTAL           8          2

NUMBER OF SECTORS USED BY ALL FILES = 222
NUMBER OF FREE SECTORS WITHIN ALL FILES = 45
NUMBER OF SECTORS ALLOCATED TO ALL FILES = 267

```

Figure 18. Summary Screen



## Disk Dump Utility

**Purpose:** Displays or prints the contents of all or part of a specified disk or disk file.

**Use:** Programmers use Disk Dump to help trace the source of program errors or to confirm the contents of a data file. For example, if the results from a sample program you have developed are wrong, a Disk Dump could be used to determine if the program is using the correct data.

### Procedures:

- There are three options for Disk Dump output formats: Horizontal, Vertical, and Data Structure. Horizontal and Vertical options print the two-digit ASCII Hexadecimal value of each byte and the corresponding characters. The Horizontal has 6 sectors per page and the Vertical 3 sectors per page. The Data Structure option is valid only for data files whose records were written using DATASAVE DC or DATASAVE DA. The Vertical requires 8 1/2 x 11 paper whereas the other two require 11 x 14 paper.
- There are also three ways in which the output can be displayed: printer, CRT display, and user-interactive display. Each display option has its own restrictions. For example, the CRT can only display Vertical formats. In user-interactive display, the user views the contents of the current sector and may position it forward or backward from one to five sectors.

**Example:** This is a dump of the program that was used to demonstrate the Compression Utility. It is clear from these examples why the horizontal dump cannot be output on the CRT and why it takes enlarged 11 x 14 paper on the printer.



### File Status Report Utility

**Purpose:** Closes or displays the status of specified multiplexed/multistation data files for one or all stations.

**Use:** This utility is designed especially for data files that may be accessed by multiplexed CPUs or multistations. It lets the user know which files are and are not available to multiplexed/multistation systems.

#### Procedures:

- The access status information that is printed is file name and one of the following:
  - Not opened - when file is not opened to any station;
  - Not a multiplexed file - when the file is not a multiplexed/multistation file;
  - Station number and respective access modes - when the file is open to one or more stations.
- MODE options are All, Part, Range, or Indirect.
- There are 4 output options, one of which is to close the data file for all possible station numbers.

Example:

```
ENTER OPTION
-
FILE STATUS REPORT ( STATION # 1 )
1 INPUT ADDRESS - B10      3 OUTPUT OPTION - LIST ANY
2 MODE - ALL              4 STATION # - 0
                          5 OUTPUT DEVICE - CRT
6 SAVE DEFAULTS          '15 RETURN TO CHANGE PARAMETERS

1 - CLOSE FILE           3 - LIST STATUS MUX'D FILE
2 - LIST STATUS ANY FILE 4 - LIST STATUS OPEN FILE
```

Figure 21. File Status Report Screen with Output Options Displayed

### Program Compare Utility

**Purpose:** Compares the text of a pair of specified program files on a line number-by-line number basis.

**Use:** This utility allows a detailed comparison of program files when moving or copying programs. It is an excellent safeguard guaranteeing that programs are exactly the same, and an easy way to determine how programs differ.

**Procedures:**

- A message is printed when:
  - A statement line number exists in only one program file;
  - The line numbers are the same but the code differs;

- One program ends and the other continues, and
- Both programs are identical.
- Remark statements and blanks are ignored in the comparison. Thus, the program only detects significant differences between programs.

Example:

```
ENTER MODE

PROGRAM COMPARE ( STATION # 1 )

1 INPUT ADDRESS ONE - B20      3 MODE - INDIRECT
2 INPUT ADDRESS TWO - 320     4 OUTPUT DEVICE - CRT
                               5 ERROR LIMIT - 0

6 SAVE DEFAULTS                '15 RETURN TO CHANGE PARAMETERS

1 - ALL      3 - RANGE
2 - PART    4 - INDIRECT
```

Figure 22. Program Compare Messages  
with MODE Options Displayed

### Reconstruct Disk Index Utility

**Purpose:** Recovers a disk catalog whose index has been accidentally scratched.

**Use:** This utility is a recovery aid and should be used only when the catalog index has been accidentally destroyed. When the catalog index has been scratched, the Reconstruct Disk Index program greatly facilitates the creation of a new index.

**Procedures:**

- The utility searches the specified disk for file control sectors and reconstructs the catalog index from this information.

The operation of this program can take a considerable amount of time, especially when numerous scratched files are on the disk.

- Program file names can be recovered by the utility but data file names cannot; thus, arbitrary file names are generated for data files.

**Example:**

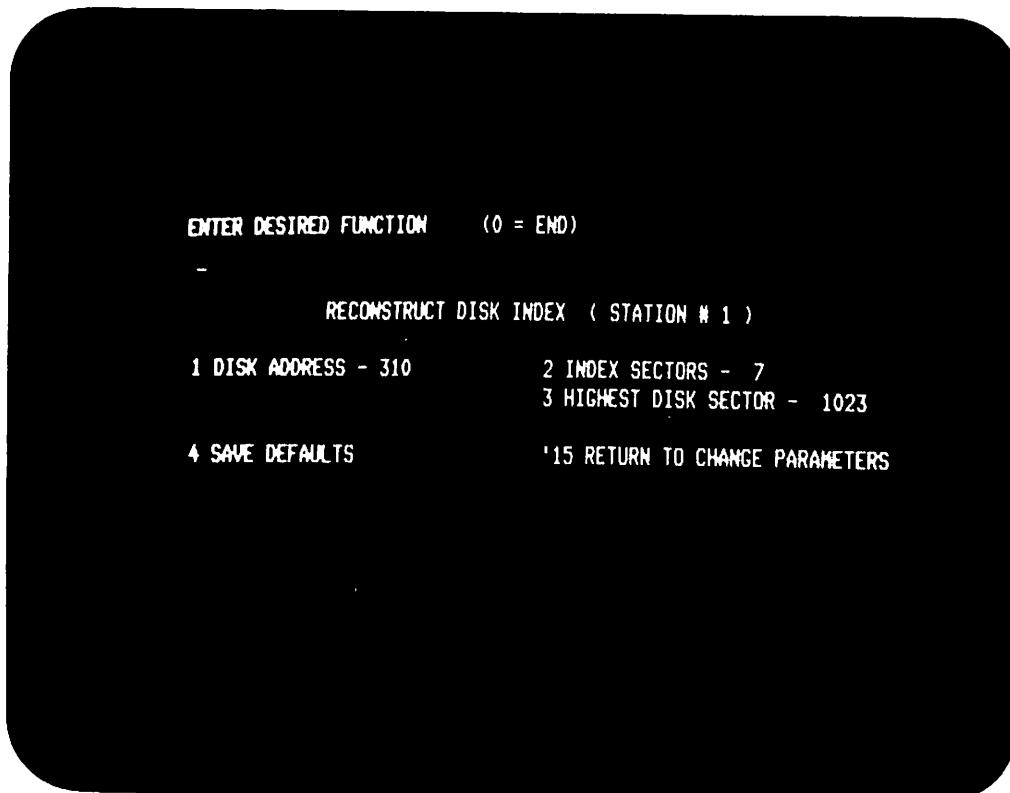


Figure 23. Reconstruct Disk Index Screen

**Alter Disk Index Utility (ISS-5 Release Only)**

**Purpose:** Optionally displays or changes the contents of a disk's catalog index and renames or changes the status of files. A file may be renamed (which automatically makes the file status active), a scratched file may be activated, an active file may be scratched, a file's usage parameters may be displayed, and the last file on a disk may be removed.

**Use:** It is important to keep the disk catalog index as current as possible so that it accurately reflects the disk contents. This utility allows for quick and easy updating of the catalog index.

**Procedures:**

- A backup copy of the disk (to be used with this utility) must be made.
- Special function keys are used to perform the various possible operations in this utility as well as to move the list of file entries forward or backward.

## INTEGRATED SUPPORT SYSTEM

---

**Example:** This is the screen that calls for a disk address so that the proper catalog index can be altered. When the index is displayed, the SF keys allow the user to take the desired action.

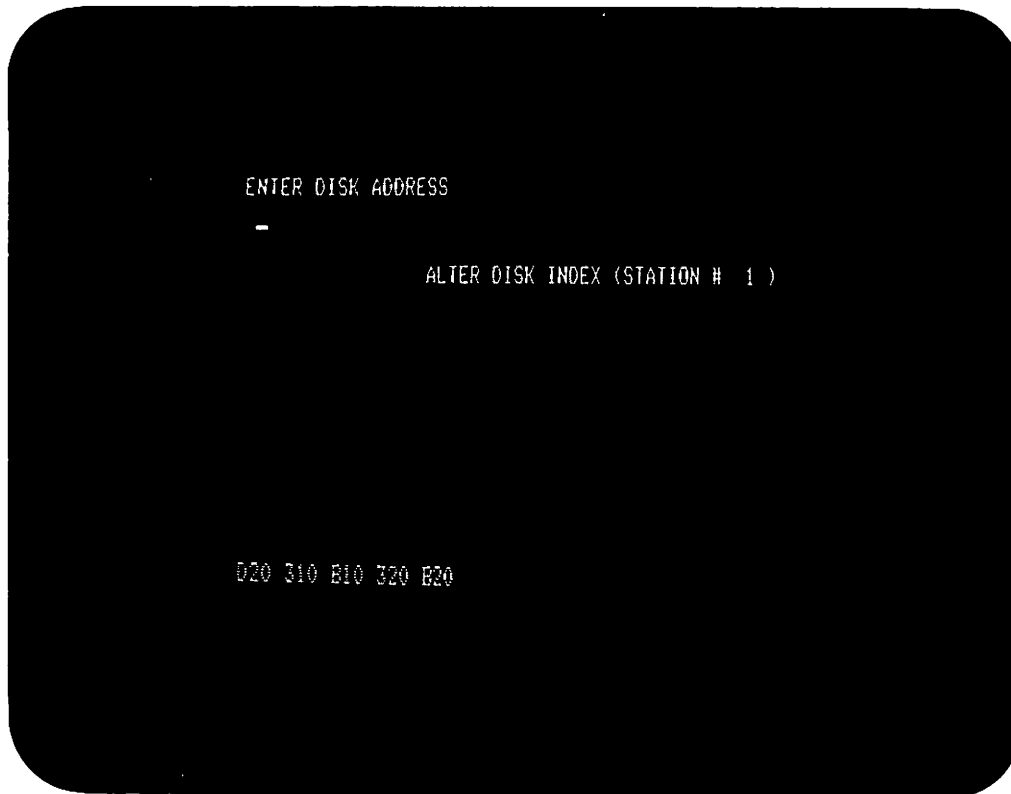


Figure 24. Enter Disk Address Screen

### Summary

In summary, the eleven ISS Utilities are all user-controlled programs which can assist in four types of jobs: Copy Function, Programming Functions, Catalog Index Functions, and Special Purpose Functions. The ISS Utilities package is a proven, comprehensive support system which provides economical and productive service to 2200 Series product line users.



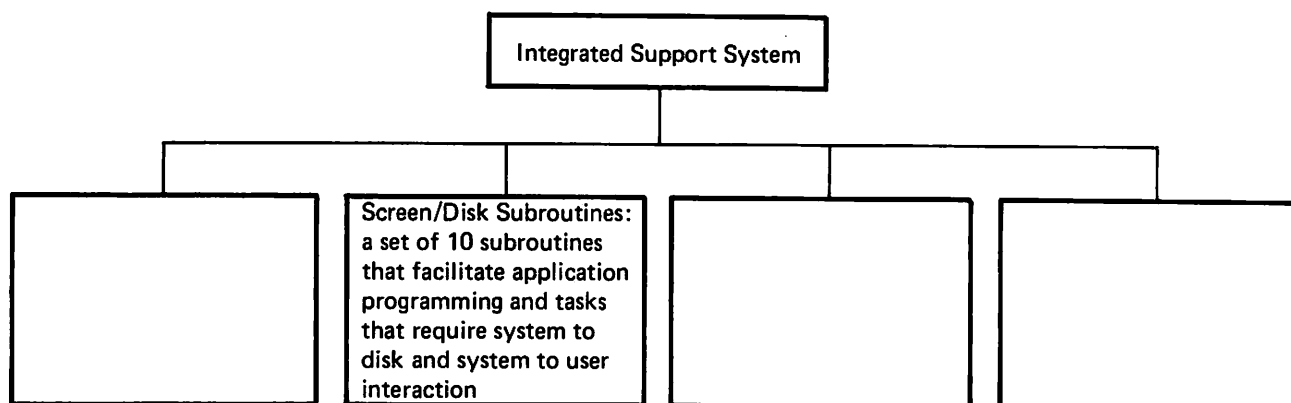
REVIEW QUESTIONS

1. Which Utility would you recommend if a customer wanted to move a data file from one disk to another and also insure that no errors had arisen because of the transfer?
  - a. Move/Validate
  - b. Reference File Utility
  - c. List/Validate
  - d. Copy/Verify
  
2. A programmer discovered that a variable was being used incorrectly in a program. Since it was a very long and complicated program and the variable was used more than once, the programmer wanted to use a utility to locate all the places where the variable was used. Which utility would you recommend?
  - a. Create Reference File
  - b. List/Cross Reference
  - c. Sort Disk Catalog
  - d. Program Compare
  
3. A disk catalog index has been accidentally scratched. What would you do?
  - a. Try Alter Disk Index
  - b. Try Sort Disk Catalog
  - c. Try Reconstruct Disk Index
  - d. Try Disk Dump Utility
  
4. A program has been written and documented with a number of remarks and ample spacing between statements. The programmer wants to store the program in the most efficient way possible. What should be done?
  - a. Compression Utility
  - b. Disk Dump Utility
  - c. Erase Unnecessary Verbage subroutine
  - d. Decompression Utility

ANSWERS

1. d. The Copy/Verify utility can accomplish this task.
2. b. The List/Cross Reference utility will provide this information. It will also provide the line number for each subroutine call.
3. c. The more constructive activity is to run the Reconstruct Disk Index utility.
4. a. The Compression utility will eliminate most remarks and extra spaces.

SCREEN/DISK SUBROUTINES



The second major component of the ISS package is Screen/Disk Subroutines which are a great help to the application programmer. Frequently used routines are available in one package; therefore, many repetitious, detailed programming tasks can be eliminated. The subroutines provide a simple interface between application programs and a wide range of potentially complex disk-related and operator-related tasks.

The Screen/Disk Subroutines do not overlap and it is possible to use all subroutines in the same program. Each subroutine can either be selected as global (i.e., available to all partitions within a bank) or local. Each Screen/Disk Subroutine is marked by the DEFFN statement. In the application program, the GOSUB statement calls the designated subroutine.

There are three separate menus for these subroutines: Screen Routines, Disk Routines, and Translation Tables. The Screen Routines menu appears when the Screen/Disk Subroutines option is selected from the ISS System Menu. SF Key 16 is pressed to display the next succeeding menu (i.e., screen routines to disk routines). Routines are selected by pressing the SF Key listed next to the routine. When the user chooses a subroutine from one of the three menus, an asterisk appears opposite the routine selected.

The menu for the Screen Routines is shown in Figure 25. In the next figure, the asterisk appears by the three routines that have been chosen by the user. These routines are then loaded into memory and saved on the specified disk with a user specified file name. This same general procedure is used for disk subroutines and the translation tables.

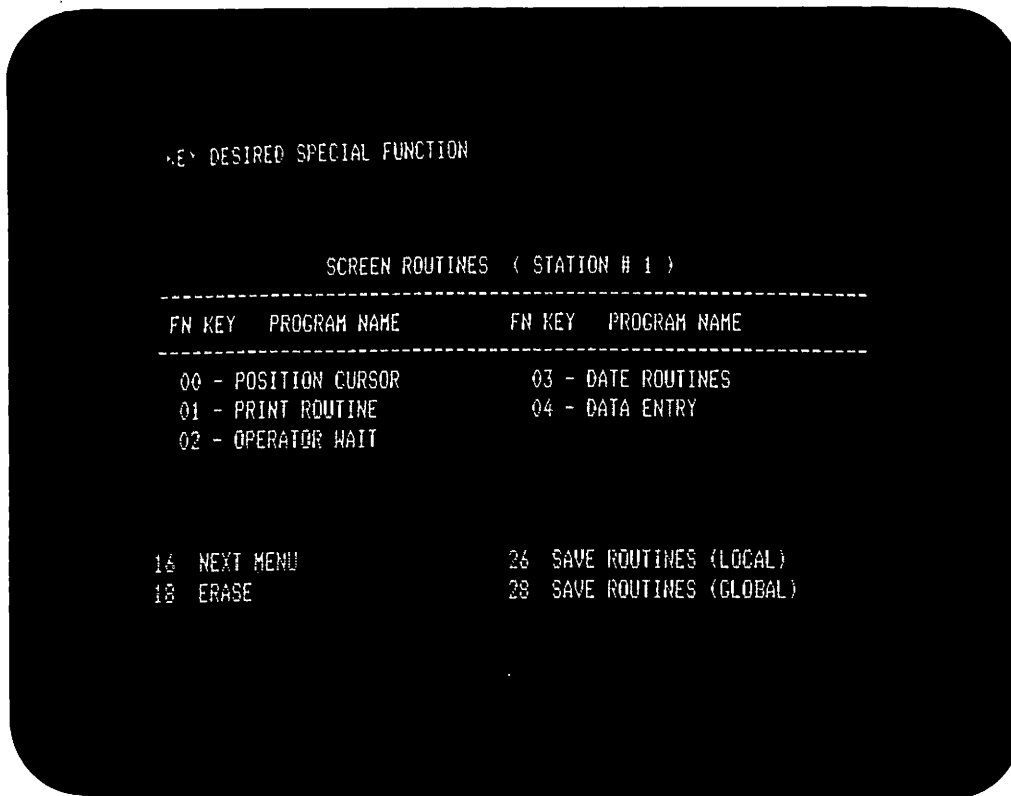


Figure 25. Screen Routines Menu

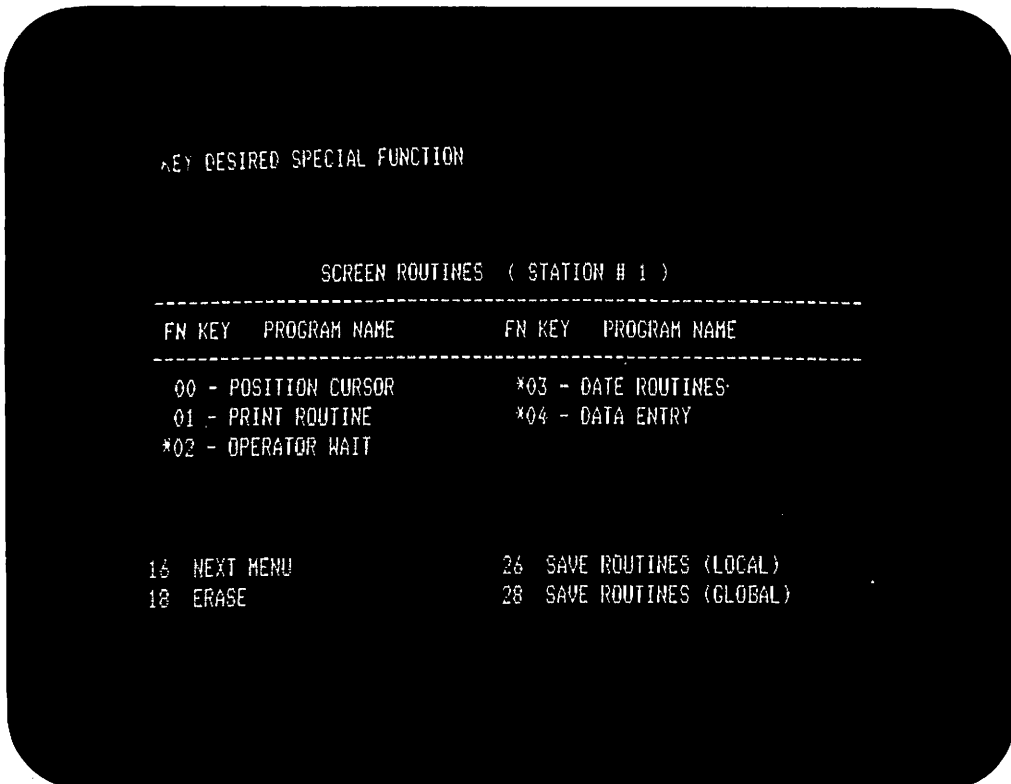


Figure 26. Subroutines Chosen from Screen Routine Menu

The three Screen Routines that will be discussed are Data Entry, Date Routines, and Operator Wait. These are the three most widely used Screen Routines.

#### Data Entry Subroutine (DEFFN' 200)

**Purpose:** Validates keyboard entries as numeric or alphanumeric input, checking each for its value and length.

**Use:** This screen subroutine is a valuable way to reduce keyboard errors. It is especially useful when the operator has a great many data entries to make. If, for example, the data to be entered are test scores that range from 1 to 99, the data entry routine can be used to send an error message if the score 149 is accidentally entered. Without this error message, the 149 score could go undetected, and perhaps require all the data to be re-entered.

#### Procedures:

- There are a number of checks performed by this routine, including:
  - Does the numeric entry conform to the minimum and maximum ranges?
  - Does the numeric entry have the appropriate number of digits to the right and left of the decimal point?
  - Does the alphanumeric entry have the proper length?
  - Does the alphanumeric entry conform to the proper limits?
- Transfer to the data entry subroutine is by a GOSUB' 200 statement with L\$, H\$, Ll, Rl, P\$, and T parameters specified, where:
  - L\$ = lowest acceptable entry
  - H\$ = highest acceptable entry
  - Ll = maximum number of characters
  - Rl = maximum number of characters to right of decimal
  - P\$ = an operator prompt
  - T = type of entry

#### Date Routines (DEFFN' 220,221,223,224,225)

**Purpose:** Allows the entry and use of dates in Julian and Gregorian forms. Gregorian dates are MM/DD/YY. Julian format is YY/DDD where DDD ranges from 1 to 365.

## INTEGRATED SUPPORT SYSTEM

---

**Use:** Gregorian dates are commonly used to record single events or transactions (e.g., date in order arrived or when the bill was sent). Julian dates are important to calculate, for example, aging in accounts receivable or the number of days between two events.

**Procedures:**

A listing of the various date routines is presented in Table 4.

Table 4. Summary of Date Routines

Name	DEFFN Number	Input	Output	Entered Via	Comments
Enter Date — Gregorian	220	Gregorian date	Gregorian and Julian date	GOSUB 220	Operator prompt must be specified
Convert Date — Gregorian to Julian	221	Gregorian date	Gregorian and Julian date	GOSUB 221	
Enter Date — Julian	222	Julian date	Julian and Gregorian date	GOSUB 222	Operator prompt must be specified
Convert Date — Julian to Gregorian	223	Julian date	Julian and Gregorian date	GOSUB 223	
Convert Julian date to proper format	224	Julian date	Julian date in proper form	GOSUB 224	If for example 72367 is entered, 73002 is the proper Julian date
Calculate days between two dates	225	Two Julian	Number of days between the two dates	GOSUB 225	

Operator Wait Subroutine (DEFFN' 254)

**Purpose:** Allows a program halt that can only be corrected by a positive action taken by the operator.

**Use:** In some programs it is valuable to build in a stop in the action. This is usually done to permit verification of a long string of data or rechecking before an important program action, such as mounting a disk or changing printer forms.

**Procedures:**

- This routine causes the following message to be displayed "KEY RETURN (EXEC) TO RESUME?"
- This routine is entered by the GOSUB' 254 statement.

## DISK ROUTINES

There are eight disk routines that can be accessed through the Screen/Disk subroutines. (The Disk Routine menu is presented in Figure 27.) Each assists the application programmer in performing common disk related functions. The six most widely used subroutines and the ones that are discussed in this module are:

- Search Index
- Allocate Data File Space
- Free Unused Sectors
- Limits Next
- Open/Close Input/Output
- Multiplexed/Multistation File Open/End/Close

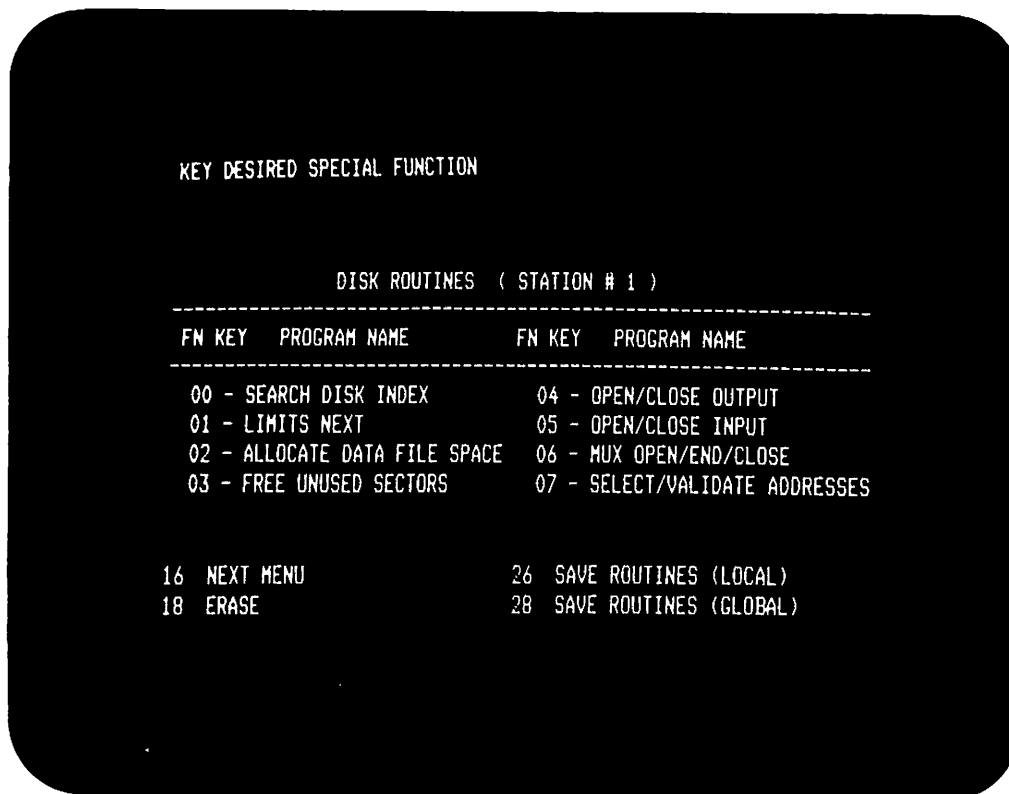


Figure 27. Disk Routine Menu



**Search Index Subroutine (DEFFN' 229)**

**Purpose:** Searches a disk catalog index for a file name and indicates the status and type of file.

**Use:** It is frequently important to be able to ascertain quickly if a file is active, scratched or non-existent, or if it is a data or program file.

**Procedures:**

- It is now recommended to use the LIMITS statement to accomplish the same purpose as the Search Index subroutine. This subroutine is included in ISS-5 primarily for support of older systems.

**Allocate Data File Space Subroutine (DEFFN' 228)**

**Purpose:** Opens and allocates file size for a new data file. Assigns the new file to the available sectors between the current end of cataloged files and the end of the cataloged area.

**Use:** Whenever a new file is opened, it is important to allocate it to the proper position in the catalog. This subroutine accomplishes this task efficiently.

**Procedures:**

When this subroutine is used, there are three conditions under which a new file will not be opened:

- If the file name is the same as an indexed, scratched file,
- If the file name is the same as an indexed, active file, and
- If there are insufficient sectors in the cataloged area.

**Free Unused Sectors Subroutine (DEFFN' 227)**

**Purpose:** Examines a specified file in a disk catalog area and reduces the number of sectors, when possible.

**Use:** There are occasions when a file has a large number of unused sectors, especially when it has been allocated by DEFFN' 228 (described above). This situation can result from an exaggerated initial estimate of file size or other changes that influence the amount of

information to be stored or the number of records in the file. This program can shrink the actual file and free the unused sectors for other use when the file in question is last on the disk.

Procedures:

- This subroutine reallocates those sectors between the END trailer record and the end of the file (extra sectors) as free sectors available to be assigned to other files.
- The initial file must have been ended with a DATASAVE DC END statement or Disk subroutine equivalent.
- When this subroutine is executed on a file which lacks an END trailer, the file is destroyed.

Limits Next Subroutine (DEFFN' 226)

Purpose: Returns the names, status, and type of each file in index sector sequence.

Use: After constant use, the disk catalog index can become difficult to reference. This program orders the file in sector sequence and provides other valuable information on each file. This can be useful if some processing is to be done on all or most files in a disk.

Procedures:

- The LIST DC statement also places each file in sector sequence order.
- The LIMITS NEXT subroutine is accessed by:

```
SELECT #F/XYX  
GOSUB' 226 (F,N$)
```

where:

F = File disk number  
N\$ = File name the sequence will begin at  
XYX = Disk device address

**Open/Close Input/Output Subroutines (DEFFN' 240, 241, 250, 251)**

**Purpose:** Provides conventions and routines for assigning and processing creation dates, volume numbers, and recycling periods for data files by utilizing specialized header and trailer records.

**Use:** Data files often have time or volume limitations. For example it may be, that a file when initially opened should have a specified life span. During this life span, data can be entered, but data will not be accepted following this period. This program allows for both time and volume limitations.

**Procedures:**

- The single sector software header and trailer records produced by this subroutine contain the following fields:

FIELD	TYPE	CONTENTS
1	Alphanumeric	HDR - indicates header EDF - indicates end of file EOR - indicates end of volume
2	Alphanumeric	file name
3	Numeric	creation date (Julian format)
4	Numeric	number of days to retain file
5	Numeric	volume number

**Multiplexed/Multistation File Open/End/Close Subroutines (DEFFN' 217, 218, 219)**

**Purpose:** Controls multiple station access to specified data files and provides password protection.

**Use:** When data files can be accessed by more than one user, a whole series of questions arise, including: Should other stations have access to this information? If so, how should the information be shared? These subroutines are designed to provide the user with several options for sharing data files.

**Procedures:**

- For multistation and multiplexed systems, these subroutines must be used instead of the DATALOAD DC OPEN and DATA DC OPEN, END and CLOSE statements.

- The OPEN subroutine allows a multiplexed/multistationed file to be opened, the access mode defined, and a password, if any, created. The four possible access modes are inquiry, read only, shared, and exclusive. For detailed specifications of these access modes and subroutine argument formats, refer to the ISS User Manual.

Translation Tables

The third component of Screen/Disk Subroutines is the Translation Tables. There are two routines which translate EBCDIC to ASCII and ASCII to EBCDIC. The menu for the Translation Tables is shown in Figure 28. Refer to the treatment of telecommunications in Module 4 of the "Introduction to 2200 Hardware Course" for the use and characteristics of both EBCDIC and ASCII.

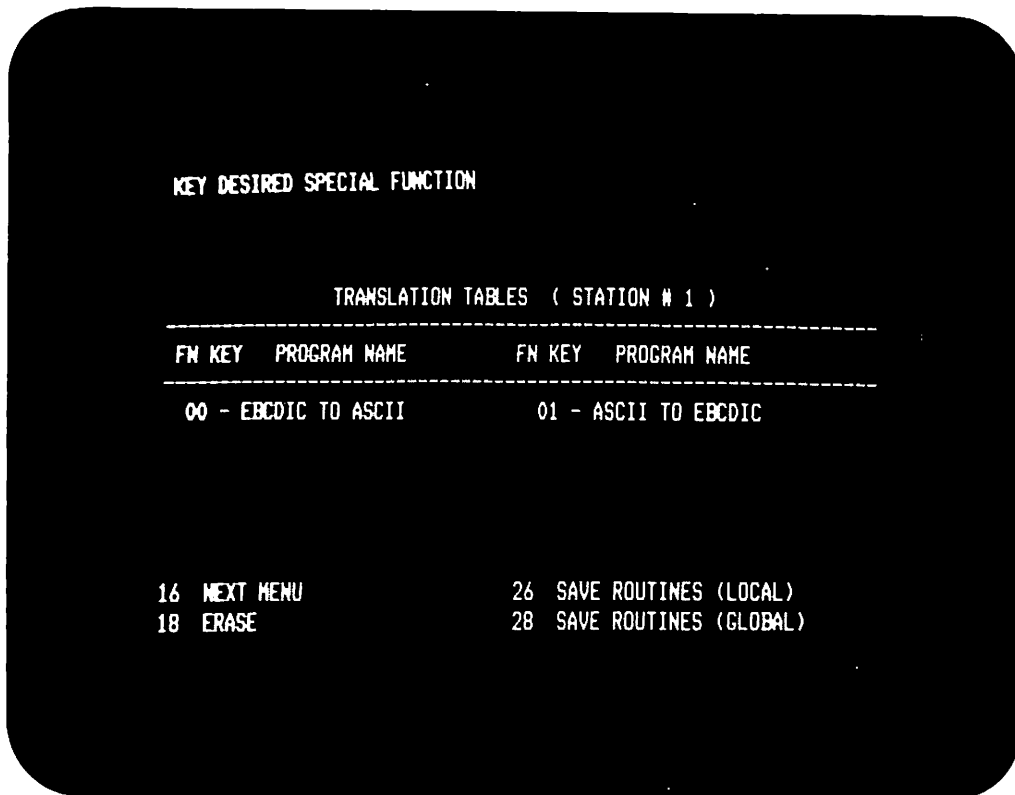


Figure 28. Translation Table Menu

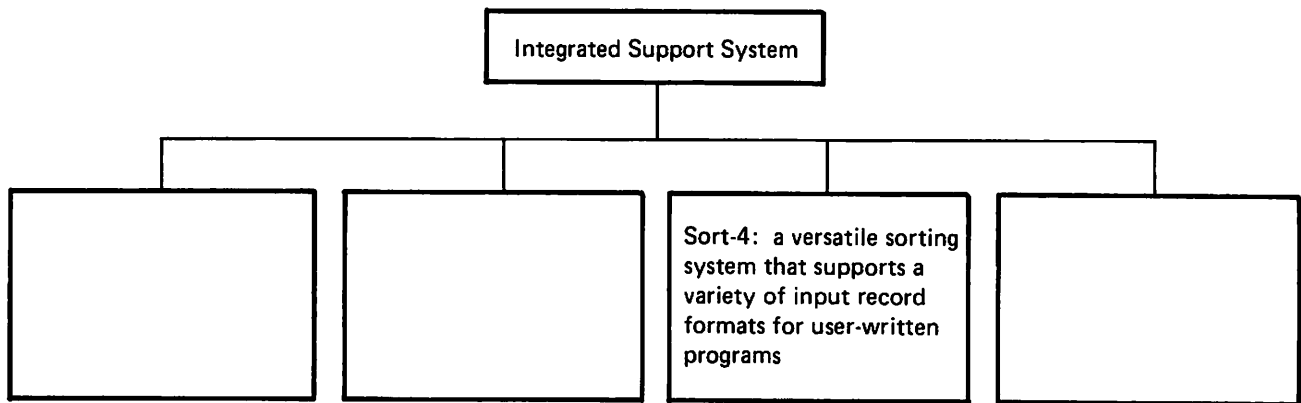
**Purpose:** Assigns specific sets of hexadecimal codes to an alphanumeric array for translating EBCDIC to ASCII (DEFFN' 202). These subroutines do not actually accomplish the translation; they merely initialize the array. The BASIC-2 statement \$TRAN performs the translation.

**Use:** In any telecommunications protocol, code is an important consideration. And in communicating between or within extended computer systems, it is often necessary to switch from one code to another. The translation tables allow these changes to be made efficiently and effectively.

**Summary**

The general purpose of these screen/disk subroutines is to assist the application programmer in a wide range of tasks requiring system-to-disk and system-to-user interaction. These subroutines do not overlap and can all be loaded at the same time. There are three main components of Screen/Disk Subroutines: Screen Routines, Disk Routines, and Translation Tables.

**SORT-4**



SORT-4 is an ISS sort program which is available to the application programmer. Since sorting is a fairly common procedure, a programmer could conceivably have to write a separate routine every time a sequencing procedure is needed. ISS not only provides a ready-made sort program but SORT-4 is an extremely versatile, powerful, and efficient tool for the application programmer. The SORT-4 component of ISS allows sorting to be economically and effectively accomplished from any user written program.

SORT-4 is designed to sort records in a specified input file into a reordered sequence in the output file. The parameters for the sort and the SORT-4 software are loaded by the user-written program. The lengthy operator-screen dialogue often required for the entry of parameters in many sorts is greatly reduced in SORT-4. Although SORT-4 requires little operator attention, it is usually run in the foreground mode in a partitioned system. Screen displays are included and, therefore, the foreground mode is required unless special coding is entered in the set-up module which permits background operation. For a 2200 VP, at least 16K of memory is required; from 9K to 12K of memory is needed for an MVP partition.

**Summary Of Sort-4 Features**

1. **TYPES OF SORTS** - The user may specify any one of three sorts: a full record output sort, a key sort (full record output), and a tag sort. Both the key sort and full record sort provide output records in exactly the same format (although in different sequence) as their input record

counterparts. A tag sort will only output pointers to the original input records; the user can access the sorted records without having to move them to a separate output file.

2. INPUT FILE FORMATS - Six different file formats are accepted by SORT-4. These formats are the following:

- An ordinary cataloged data file,
- A BAS-1 data file,
- A data file opened and closed with ISS Open and Close Output/Input subroutines,
- A KFAM-3 file,
- A KFAM-4 file, and
- KFAM-5 or KFAM-7 files.

3. INPUT RECORD FORMATS - Four different input record formats are supported by SORT-4:

- Packed arrays - Where array type blocking is packed for writing on disk.
- Contiguous packed records - Where each individual record is packed into a contiguous space within an alphanumeric array.
- Variable length records - Packed into an alphanumeric array with either a one-byte length indicator (block size up to 256) or a two-byte length indicator (greater than 256). Telecommunication files are supported by a separate variable length record format.
- Individual alphanumeric fields - In records written in unpacked format, blocked or unblocked may contain packed subfields.

4. SORT KEY - The sort key\* can contain up to 10 fields. Sort key fields may be alphanumeric or numeric, but their total length must not exceed 64 bytes. Sort order may be specified as ascending or descending for each field.

---

\* A key usually pertains to a few important fields in a record which are used for sorting or other access operations. Name or an identification number are often-used key fields for personnel records.



5. SIZE OF INPUT RECORD - An input record may contain up to 255 fields with each array element counting as one field.
6. SORT-4 MODULES - The SORT-4 program is actually comprised of a series of modules, each with its own distinct purpose. These modules accomplish such functions as the overlay of SORT-4, the processing of sort record and key formats, and the generation of code for the specified sort.

### Sort Comparisons

Before concluding this treatment of SORT-4, it is necessary to present more information on the three types of sorts. These sorts are the essence of SORT-4 and you must know the features and relative advantages of each in order to answer questions that may arise.

The key sort extracts the sort key from the input record, packs it in sort format, and appends it to a 4-byte pointer to the original input record. The record processed by the SORT contains only the sort key and pointer. When all sort records have been sorted, there is a final pass which reads the sort records in sorted sequence, uses the pointer to locate the original input record, reads the input record, and copies it to an output file in sorted sequence. The key sort is very fast and efficient through the input, sort, and merge phases, but slows down considerably in the last pass because it must read the entire input file again in random record sequence.

The full record sort packs the entire input records into a maximum of five "buckets" of 64 bytes each, where the first bucket is the sort key. On the last pass of the merge, sort records are unpacked and written in sequential order in the output file. The full record sort is generally slower than the key sort during code generation, reformatting input, sorting, and merging because there are more fields to be defined and moved. The full record sort, however, gains time because it does not read the input file in random sequence in the last pass.

Many factors influence whether a full record or key sort is more efficient for a particular application. Among the relevant factors are record length, sort key length, amount of available memory, limitations on the full record sort, proportion of key size to record size, and input blocking. When no sort type is specified in setting up SORT-4, the program cycles through these decisions and selects a type of sort (i.e., full record or key sort). This method does not, however, always determine the fastest way of sorting a particular file; it is often valuable to experiment with both types.

**INTEGRATED SUPPORT SYSTEM**

The third type of sort is the tag sort. It operates like a key sort except that the output provides only the pointers to the original input records and not the full records themselves. A user program can then access the input records in sorted order without having to move the input records to a separate output file. This feature eliminates the output pass required in the key sort and reduces the size of the output file considerably.

The three types of sorts can be compared in the following figure.

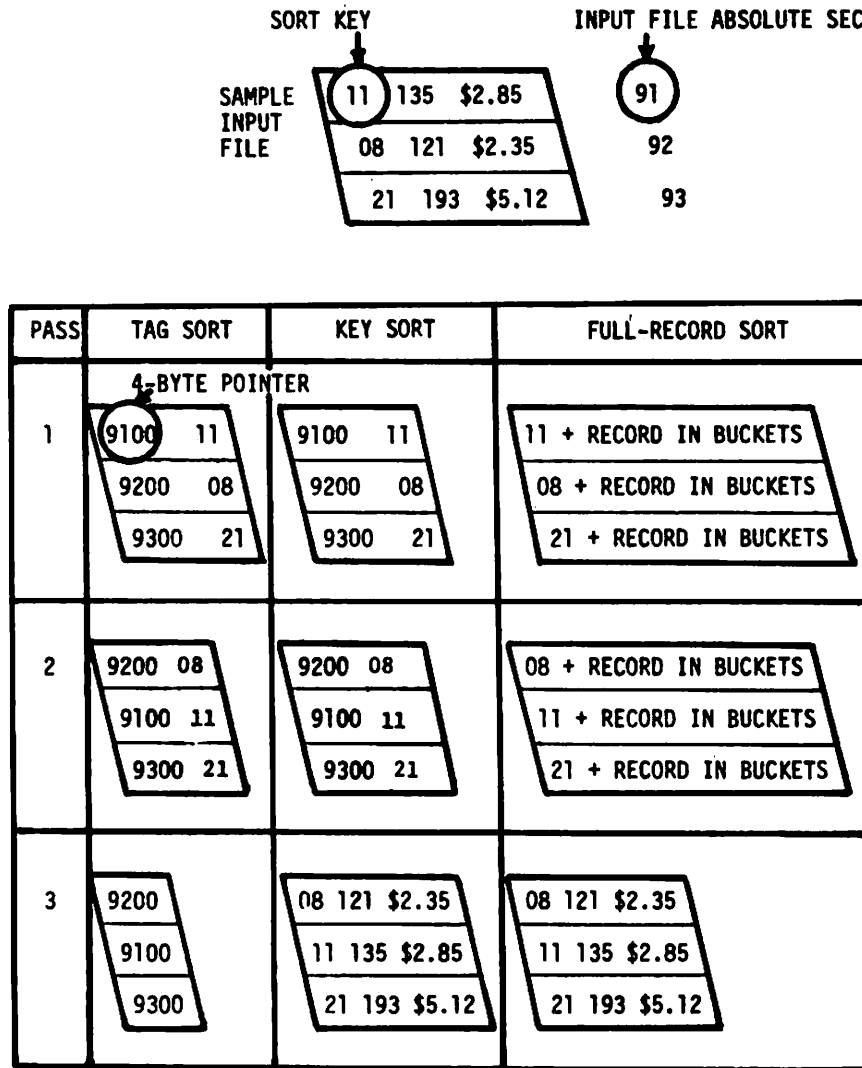


Figure 29. SORT-4 Sample Operation on Input Records

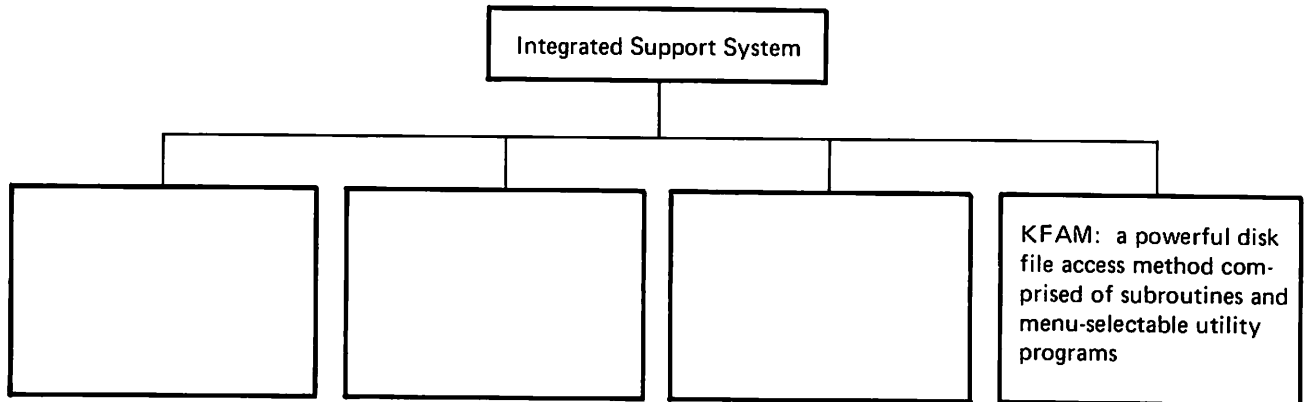
Figure 29 illustrates the basic content and arrangement of three sample input records during each pass of the sort, and for the three types of sorts. Assume the three unblocked input records are located at (absolute) sectors 91, 92, and 93 respectively and are sorted into ascending order by their sort keys. Notice

that during passes 1 and 2, the tag sort and key sort are identical. After pass 3, the key sort uses the pass 2 pointers to read the input file records and copy the records in sorted sequence to the output file. The full-record sort, however, carries the entire record for the duration of the sort (in buckets), whereas the key sort carries only pointers (and sort key) through passes 1 and 2. Note that the output of the key sort and full-record sort are identical, and differ greatly from the output of the tag sort.

#### Summary

SORT-4 is a versatile and powerful sort capability. Three types of sorts are available: the full record sort, a key sort, and a tag sort. Up to ten ascending or descending fields in each record comprise the sort key which determines the output record order. SORT-4 supports a wide variety of input file and record formats. These formats are established in a short setup program; the setup program also significantly reduces the amount of user-screen interaction which slows down many sort programs. Because SORT-4 offers the user an efficient and effective system for performing sorting operations, it is an important component of the total ISS package.

KFAM (KEY FILE ACCESS METHOD)



KFAM is a disk access method and, like all such methods, it provides a means of transferring data between memory and a mass storage device such as a disk or diskette. Effective information retrieval requires a fast and efficient disk access system. A computer system must have a powerful access method; ISS provides 2200 Series computers with KFAM.

It is important to distinguish KFAM from file catalog procedures available in Wang's BASIC-2 language. BASIC-2 includes a group of disk operation statements known as Automatic File Catalog statements which create and maintain a catalog or index of the files stored on the disk. The File Catalog keeps track of the name and location of files, but it is not concerned with the location of individual records within a file. For example, a disk may have an employee file named PEPL, an accounts file named ACRL, and a payroll file (PAYR). The Automatic File Catalog system keeps track of the location of these files. The PEPL file consists of 250 records, the ACRL has 500 records, and PAYR also has 250 records; KFAM is the system that locates individual records within each of these files. Without KFAM the individual records in these three files would be much more difficult to locate and access time would be slower.

A KFAM file actually consists of two files: 1) the file containing the data records is called a "User File" (such as PEPL, ACRL or PAYR); 2) the file containing an index for quickly locating specific records in the "User File" is called a "Key File". Within each data record in the User File is a key such as a social security or other unique identification number. The Key File contains system information that is used internally by KFAM, in addition to the key and its corresponding address in the User File.

KFAM subroutines do all the work of searching and updating the Key File. There are KFAM subroutines to find records in a random ascending or descending key sequence, to delete records,

## INTEGRATED SUPPORT SYSTEM

---

to find a location for a new record, and to add a new key to the Key File. Therefore, the programmer who uses KFAM does not need to know the structure of the Key File; KFAM subroutines carry out all the necessary operations on the Key File.

It is important to realize that KFAM--like the entire ISS package--has undergone extensive testing and improvement. The early versions of KFAM (KFAM 1-4) were designed to operate on single-user 2200 Series computers. The two latest releases (KFAM 5 and 7) are for disk multiplexed and multi-user systems such as the 2200MVP. KFAM-7 MVP actually comes in two versions: disk multiplex and 2200MVP. KFAM-5 is available on ISS-3 and KFAM-7 is on ISS-5.

A typical KFAM-7 MVP configuration is presented in Table 5. The KFAM subroutines are located in a global partition and require about 9K of memory. In each partition an additional 1 to 2K is needed for KFAM variables. On an MVP system, 30 KFAM-7 files can be opened.

Table 5. KFAM MVP Configuration

Station #3 Application plus KFAM Variables
Station #2 Application plus KFAM Variables
Station #1 Application plus KFAM Variables
Global Partition KFAM Subroutines
3K Overhead

Components of KFAM

KFAM is comprised of three different functional components: Set-up Utilities, KFAM Subroutines, and Supplementary Maintenance Utility Programs. Each of these components will be discussed in turn.

SET-UP UTILITIES: These are standalone utility programs that are used to initialize a new KFAM File and to create a Key File for an already existing User File. Two important Set-up Utilities are INITIALIZE KFAM FILE and KEY FILE CREATION.

- INITIALIZE KFAM FILE - This utility must be run whenever a new KFAM file is established. It calculates the required size of the Key File and it saves the necessary information about the User File in the Key File.
- KEY FILE CREATION - If a previously cataloged User File contains data, this utility can be run after INITIALIZE KFAM FILE. It reads the User File and creates an entry in the Key File for each record.

KFAM SUBROUTINES: These subroutines are designed to simplify the file access and maintenance operations most frequently performed on KFAM files. Among the tasks performed by the KFAM subroutines are adding new records to a file, deleting existing records, locating existing records, and accessing a file's records in ascending or descending order. The subroutines never alter the data in the User File. They operate on the data in the Key File. The specific KFAM subroutines that are available are the following:

General Purpose Subroutines

OPEN	DEFFN 230	Open specified User File and companion Key File.
CLOSE	DEFFN 239	Close User File and companion Key File.
RE-OPEN	DEFFN 213	Change the access mode of a currently open KFAM File.
WRITE RECOVERY INFORMATION	DEFFN 214	Write current file END record at end of active data in User File, and write recovery information in the next-to-last sector (both of which would otherwise only occur when a file is closed) without closing the file.

Random Access Subroutine

FINDOLD	DEFFN 232	Locate specified key in the Key File; set User File Current Sector Address to record in User File with that key.
---------	-----------	--

Key Sequence Access Subroutines

FINDFIRST	DEFFN 235	Locate record with lowest key in User File; set User File Current Sector Address to that sector.
-----------	-----------	--

FINDPREVIOUS	DEFFN 212	Locate previous record in User File in logical key sequence; set User File Current Sector Address to that sector. May be executed in any situation where FINDNEXT is allowed.
--------------	-----------	---

FINDNEXT	DEFFN 237	Locate next record in User File in logical key sequence; set User File Current Sector Address to that sector.
----------	-----------	---

FINDLAST	DFFN 236	Locate record with highest key in User File; set the User File Current Sector Address to that sector.
----------	----------	---

Add and Delete Subroutines

FINDNEW	DEFFN 233	Add specified key to Key File; allocate space for a new record in the User File, and set the User File Current Sector Address to that sector. Adds one to record count.
---------	-----------	---

FINDNEW (HERE)	DEFFN 234	Add specified key to Key File; set the User File Current Sector Address to the sector where the new record is to be written. It is normally used to change the key of a deleted record; therefore, it is normally preceded by a DELETE. Adds one to the record count.
-------------------	-----------	---

DELETE	DEFFN 231	Remove specified key from Key File; set the User File Current Sector Address to the record that has the deleted key. Subtracts one from the record count.
--------	-----------	---

**Special Purpose Subroutines**

RELEASE	DEFFN 238	Allow a User File record, previously protected by one station, to be accessed by any station.
---------	-----------	---

With KFAM-7 all of these subroutines can be loaded and run in a 9K global partition. The programmer may also select just the subroutines needed for the application by the BUILD SUBROUTINE MODULE utility. This utility creates a global module with just the selected subroutines.

**SUPPLEMENTARY MAINTENANCE UTILITY PROGRAMS:** The KFAM sub-routines perform most of the file maintenance but a group of supplementary programs are included to carry out certain maintenance tasks that are occasionally required. Among the specific maintenance utilities are the following:

- **REORGANIZE UTILITIES** - When a record is deleted by using the DELETE subroutine, its key and location are removed from the Key File. Its record in the User File is not, however, removed; and if this space is not re-used, the User File can become overcrowded with deleted records. The REORGANIZE UTILITIES puts records into ascending key sequence, eliminates deleted records and then automatically constructs a new Key File. This also re-uses space left by deleted records in the Key File.
- **REALLOCATE KFAM FILE SPACE** - This utility is used with the ISS Copy/Verify utility to copy a KFAM file and increase or decrease the amount of disk space allocated to it.
- **PRINT KEY FILE** - This utility prints the complete contents of the access table and the current contents of the Key File. It can be useful as a diagnostic tool and helpful to advanced programmers who may want to examine the Key File Structure.
- **KEY FILE RECOVERY** - This utility reconstructs the Key File in case it is accidentally destroyed. The User File must be intact for this program to operate successfully.
- **KFAM CONVERSION UTILITIES** - Utility programs are provided to convert KFAM-3 and -4 files to KFAM-5 and -7 format.



## INTEGRATED SUPPORT SYSTEM

KFAM, then, is comprised of three separate components: Set-up Utilities, KFAM Subroutines, and Supplementary Maintenance Utilities. The heart of the KFAM system is the subroutines which, when incorporated into a user-written application program, can open and close KFAM files, locate random or sequential user file records, and add or delete keys in the key file. Records in KFAM files may be accessed in ascending, descending or random key sequence. KFAM has evolved since the early days of single user, 2200T based computers and today it is used on the advanced multi-user, multitask 2200 Series computers. The KFAM system will continue to be revised and improved in order to provide 2200 Series customers with a fast, efficient, and effective disk access system.

### SUMMARY

The Integrated Support System (ISS) provides a wide range of programming and utility support for the 2200 Series product line. In this module, you have read descriptions of ISS Start-up procedures and the four main parts of ISS: ISS Utilities, Screen/Disk Subroutines, SORT-4, and KFAM. Each of these parts has its own distinct function and role:

- ISS Utilities - A group of 11 utility programs that provide important disk file maintenance functions.
- Screen/Disk Subroutines - A set of 10 subroutines that greatly facilitate application programming and tasks that require system-to-disk and system-to-user interaction.
- SORT-4 - A versatile sorting system which supports a variety of input record formats for user-written programs.
- KFAM - A disk file access method comprised of subroutines and menu-selected utility programs.

The purpose of this module has been to introduce you to ISS and its specific routines and utilities. A summary listing of these components is contained in Table 6. This treatment has necessarily been limited to a discussion of purpose, use, and procedures, with an example for each program. For more detailed, comprehensive coverage, please refer to the ISS User Manual.

Table 6. Summary of ISS Components  
(DEFFN Subroutines in Parenthesis)

ISS Utilities	Screen/Disk Subroutines	Sort 4	KFAM
Copy/Verify Create Reference File List/Cross-Reference Compression Decompression Sort Disk Catalog Disk Dump File Status Report Program Compare Reconstruct Disk Index Alter-Disk Index	Screen Routines Position Cursor (248) Print Routine (242) Operator Wait (254) Data Routines (220-225) Data Entry (200) Disk Routines Search Disk Index (229) Limits Next (226) Allocate Data File Space (228) Free Unused Sectors (227) Open/Close Output (240,241) Open/Close Input (250,251) MUX Open/End/Close (217,218,219) Select/Validate Addresses (205) Translation Tables (201,202)	Full Record Sort Key Sort Tag Sort	Initialize KFAM File Build Key File Reorganize In Place Reallocate File Space Convert to KFAM-7 Print Key File Reset Access Tables Build Subroutine Module Key File Recovery KFAM Subroutines (212-214, 230-239)

REVIEW QUESTIONS

1. A programmer wants to know if a particular file can be accessed by all MVP stations. Which utility should be used to find the answer?
  - a. List/Cross Reference
  - b. File Status
  - c. Program Compare
  - d. Create Reference File
  
2. A programmer wants to make it illegal for an operator to enter a number higher than 650. What routine would you suggest?
  - a. Data Routines
  - b. Limits Next
  - c. Allocate Data File Space
  - d. Data Entry
  
3. A programmer knows that there are many vacant, unused sectors in a file. These sectors are wasted space and she wants to make them available for program or data storage. What utility should be used?
  - a. Allocate Data File Space
  - b. Limits Next
  - c. Free Unused Sectors
  - d. Data Entry
  
4. What are the three types of sorts that can be selected in SORT-4?
  - a. Small, Medium, Large
  - b. Packed, Unpacked, Array
  - c. Number, Letter, Alphanumeric
  - d. Full record, Key, Tag
  
5. Which type of sort provides the limited output of the pointers to the original input records?
  - a. Tag
  - b. Small
  - c. Unpacked
  - d. Key
  
6. A KFAM file actually consists of which two types of files?
  - a. KF and AM files
  - b. Tag and Full files
  - c. Key and User files
  - d. Sequence and Random files

7. The most important of the three components of KFAM is:
  - a. Supplementary Maintenance Utility Programs
  - b. Subroutines
  - c. Set-up Utilities
  - d. Primary Utility Programs

ANSWERS

1. b. The File Status ISS Utility will provide this information.
2. d. The Data Entry subroutine can establish limits for inputted data.
3. c.
4. d.
5. a.
6. c. A KFAM file consists of a key and user file.
7. b. The subroutines are the heart of KFAM.

**MODULE 2**

**GENERAL BUSINESS SYSTEM (GBS)**

CONTENTS

ABSTRACT.....TBS-1

OBJECTIVES.....TBS-1

MATERIALS REQUIRED.....TBS-2

DIRECTIONS FOR COMPLETION.....TBS-2

GENERAL BUSINESS SYSTEM (GBS).....TBS-3

GBS FEATURES AND BENEFITS.....TBS-6

INVOICING/ACCOUNTS RECEIVABLE GBS APPLICATION.....TBS-15  
     Application Data Files.....TBS-15  
     Application Menu and Examples.....TBS-17

ORDER ENTRY/INVENTORY CONTROL: GBS APPLICATION.....TBS-21  
     Application Data Files.....TBS-21  
     Application Menu and Examples.....TBS-22

ACCOUNTS PAYABLE, GENERAL LEDGER GBS APPLICATION.....TBS-29  
     Application Data Files.....TBS-29  
     Application Menu and Examples.....TBS-31

PAYROLL GBS APPLICATION.....TBS-36  
     Application Data Files.....TBS-37  
     Application Menu and Examples.....TBS-37

BILL OF MATERIALS GBS APPLICATION.....TBS-39  
     Application Data Files.....TBS-39  
     Application Menu and Examples.....TBS-40

APPLICATION SUMMARY.....TBS-42

GBS UTILITIES.....TBS-44

GBS FILE REORGANIZATION PROGRAMS.....TBS-45

INVENTORY MANAGEMENT SYSTEM.....TBS-45

SUMMARY.....TBS-47

REVIEW QUESTIONS.....TBS-49

ANSWERS.....TBS-52

## ABSTRACT

General Business System (GBS) has been developed by Wang to serve the business and accounting needs of 2200 Series product line users. The GBS package is licenced to independent software vendors who, in turn, modify the programs, as necessary, to fit the customer's data processing requirements. As you recall from the Introduction to 2200 Software course, GBS is not completely a turnkey or a customized system, but rather it is a combination of the two. The main body of the GBS programs is standard but parts can be modified to personalize and tailor the system.

The purpose of this module is to familiarize you with the specific features and benefits of the GBS system and its individual applications. As you will see, GBS is a large, comprehensive system: much larger than either ISS or IDEAS. And it would be inappropriate for these training materials to define every menu option, data field, or entry on a report. A series of GBS Technical, System, and User Manuals accomplish these tasks. But these training materials do provide a firm understanding of GBS and its application so that you have a sound basis for further analysis and study of GBS.

These training materials deal with such activities as order entry, inventory control, accounts receivable, accounts payable, and general ledger. Therefore, an understanding of general principles of accounting is essential. You should complete a background course in accounting if you are not familiar with general accounting principles for small businesses. Introduction to GBS (700-4186C) could also be read as a way of gaining some background information on accounting and GBS.

## OBJECTIVES

Upon successful completion of this module you will be able to:

- Define the general purpose and market for GBS.
- List specific GBS features and benefits.
- Define GBS minimum hardware requirements.
- Discuss Data File Initialization Procedures and the use of the GBS Configurator.
- Identify the five GBS applications and their dependent and independent relationships to each other.
- Describe each of the five GBS applications in terms of purpose, data files, menus, and examples of reports and displays.



## **GENERAL BUSINESS SYSTEM**

---

- Define the purpose of GBS Utilities.
- Define the purpose of GBS File Reorganization program.
- Describe the purpose and general features of the Inventory Management System.

### **MATERIALS REQUIRED**

There are no additional materials required for completion of this module.

### **DIRECTIONS FOR COMPLETION**

After completing this Module, contact the Course Administrator for the Module Test.

GENERAL BUSINESS SYSTEM (GBS)

NOTE: Technically, these training materials pertain to GBS Release 2. For ease of discussion, GBS Release 2 is simply referred to as GBS throughout this module.

Wang's GBS software is a system that is designed to meet the business needs of small- to medium-sized businesses. As you already know, GBS is an extremely popular application package and has greatly contributed to the sale of many 2200 Series product line computers. Customers are interested in total, not partial, answers to their data processing needs. GBS--together with the high performance 2200 Series hardware and Wang's commitment to provide excellent service through its associated software consultants - comprise Wang's total solution.

GBS is a package that has been designed and developed by Wang and licenced to independent software vendors. The vendor modifies the GBS package as needed and installs the package in the customer's 2200 system. The customer's direct contact is through the Wang-authorized vendor. Wang does, however, have responsibility for monitoring the vendor's performance in adapting and installing GBS. The analyst needs to be very familiar with the features and structure of GBS, not only to monitor the vendor's performance but also to be able to answer technical questions as they arise.

Wang's GBS software has the following capabilities:

- Invoicing programs create postbilling invoices that contain ship-to, billing, and relevant product information through interaction with the customer, salesman, and inventory files.
- Accounts Receivable programs automatically interface with the invoice system through the Invoice Transaction File, thereby updating the Open Item and Control Files. The Open Item File accounts for both balance forward and open item customers, and generates statements, aged trial balances, and open item reports.
- Sales Analysis reports are single-line sales reports depicting sales and cost extension figures along with profit margins for current month and year-to-date figures. They are printed by the customer, inventory, and salesman file display/print programs.
- Order Entry programs process customer orders, update open order and appropriate master files, permit adjustment and confirmation of orders, create and maintain backorders, and print shipping papers and invoices.

## GENERAL BUSINESS SYSTEM

- Inventory Control programs update the Inventory Master File with receipts or withdrawals from inventory. They also produce stock status, low stock, inactive items information, physical inventory sheets, and recommended purchase order reports.
- Bill of Materials programs provide and maintain a product structure file for manufacturing. Raw materials inventory quantities and costs can be related to finished goods inventory. The system produces Bill of Materials explosions, Where-Used listings, and gross requirements analysis.
- Accounts Payable programs perform accounts payable posting, check writing, and automatic general ledger updating.
- General Ledger programs perform general ledger posting and produce corporate financial reports including trial balance and balance sheets.
- Payroll programs calculate earnings, taxes and deductions based upon entered and/or stored data for hourly, hourly-exempt, and salaried employees. They support check, cash, and direct-deposit payment systems.

These programs are combined to form five GBS applications (Figure 1).

1. Invoicing, Accounts Receivable
2. Order Entry, Inventory Control
3. Accounts Payable, General Ledger
4. Payroll
5. Bill of Materials.

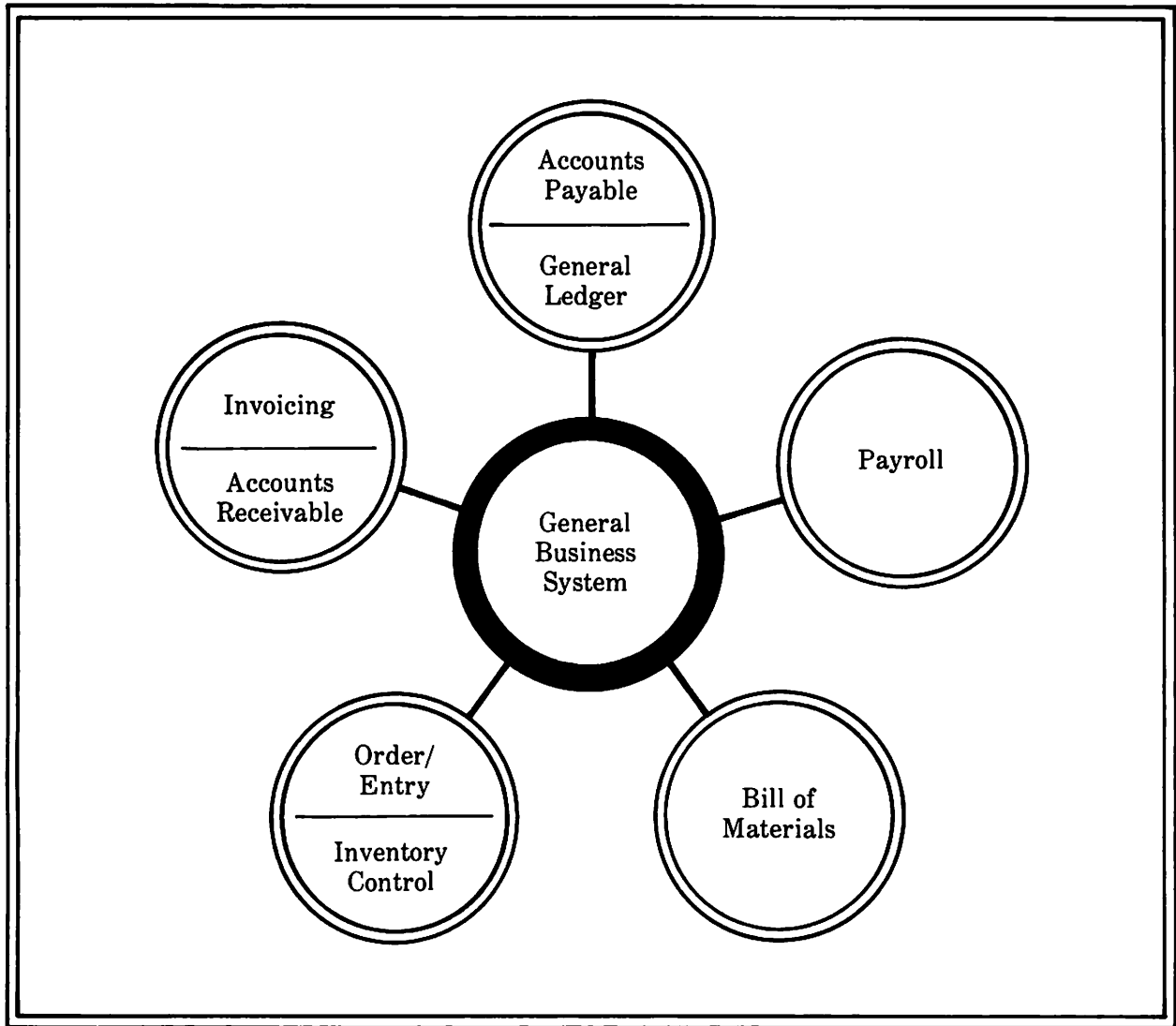


Figure 1. Five GBS Applications

## GBS FEATURES AND BENEFITS

Wang's General Business System, because it is such a comprehensive system, has a number of important features and benefits. Before discussing several specific GBS features, it is important to remember that GBS is a multiuser, disk-based, system. GBS is designed for the MVP and LVP and, therefore, can be accessed by several different operators simultaneously.\* One operator, for example, can be running Order Entry/Inventory Control while another can be updating Payroll. And because GBS is a disk-based system, critical data and programs are easy to load, unload, store and access; these operations are completed very quickly.

In addition, there are a number of more specific GBS features, including the following:

- Common utilities and subroutines are used throughout GBS for similar, repetitive tasks. GBS is therefore a modular and very efficient software package.
- KFAM-7 is used for rapid file access and handling.
- SORT-4 is included for efficient file sorting and rearrangement.
- ISS Utilities are included in the package to simplify program modifications and maintenance.
- File I/O routines are modularized to simplify program changes in read/write routines.
- Display/Print programs can be used to inspect information in data files through screen displays or hard copy print outs. These programs are very useful in problem analysis and debugging.
- Audit reports print out the changes to files as a result of file maintenance operations. This feature allows the user to trace changes that have been made in the data base.
- A series of programs accomplish data base maintenance for the purpose of efficiently adding, changing, or deleting records.
- Files are reorganized through a set of menu accessible programs. These programs serve to make the most

---

\* There are versions of GBS available for VP and SVP single-user systems.

effective use of available space by deleting inactive records and making room for the creation of new records.

- Password protection for files is provided at several levels; there is a company password and it is also possible to specify a password for each file.
- Multicompany capabilities are supported by GBS. The feature is valuable in environments such as service bureaus where information needs to be kept on a number of different companies. GBS can process information on as many as 200 different companies.
- File backup procedures are provided as a part of the standard operating system.

These are several of the most significant GBS features; others can be found in GBS documentation (such as the Data Sheet, Technical Guide, and User Manuals for each application). These features, of course, need to be translated into user benefits before they influence sales and make an impact on the customer. Perhaps all of the preceding features can be categorized under four benefits to the end user.

- Time Savings - Time is saved because GBS is easy to use and many time-consuming tasks are taken care of by the system. For example, file maintenance, reorganization, and sorting are accomplished with very little user interaction. The customer, then, saves time because of the efficient operation of the system, and the fact that less time is needed to train personnel on how to use GBS. In addition, the software vendor can deliver a GBS system to the customer in a relatively short period of time because GBS is modular and so easy to modify.
- Flexibility - Several GBS features add to the systems flexibility. One example of this flexibility is the audit reports which track changes made to the data base. Another example is the multicompany capability of GBS which greatly expands the number of companies that can utilize GBS. The final example is the modularity of the structure of GBS programs which permit easy adaptation of GBS to suit the business needs of the customer.
- Easy to Use - GBS is a menu-driven, conversational system that the first-time user can operate. Because GBS is extremely user-oriented, training time is saved, fewer operation and data entry errors are made, more confidence is developed in the system, and there is a greater likelihood that GBS will make an impact on the customer's business.

## GENERAL BUSINESS SYSTEM.

---

- A Powerful System - GBS is a powerful system because it utilizes the fast, reliable, and effective 2200 Series product line; and because, once a data base is created, the GBS programs calculate totals, formulate financial reports, print invoices, and so on. In most cases, the only action by the operator is simply a choice of options on a main application menu.

Refer to the Features/Benefits chart at the end of the "Introduction to 2200 Software" course (910-2002).

### HARDWARE REQUIREMENTS

The minimum hardware configuration for GBS is contained in Table 1.

Table 1. GBS Minimum Hardware Requirements

CPU	— 2200 MVP, LVP, VP* or SVP* with 32K
Input	— 2236 DE Interactive Terminal
Storage	— F/R Disk Drive
Printer	— Any 132 Character per line printer

In terms of CPU requirements in multiuser systems, allow 16K per bank for system overhead and global text and 16K for each terminal. GBS is also VP and SVP compatible. All subroutines and utility programs that were previously re-entrant (or global) code are now overlaid into memory at run time. Those routines affected by the operating environment of the SVP and VP are provided on a separate diskette. The minimum memory requirement for the VP and SVP versions is 32K. Since the VP version uses the multiplexed version of KFAM-7, two or more VP processors can be multiplexed to one disk drive.

---

\* There are single-user versions of GBS for use on VP or SVP systems. The full potential of GBS cannot be realized on the SVP due to limited user memory.

**DATA FILE INITIALIZATION**

As you will notice when you either work with or receive a GBS shipment from Wang, GBS is not initially on a hard disk; but rather it is contained on diskettes. GBS is delivered to the vendor in two sets: a set of source programs, and a set of compressed programs. Source programs are heavily remarked with one instruction per line, while compressed programs have no remarks and contain multistatement lines. If GBS is to be adapted, the source code diskette should probably be used. Then the programs can be compressed with the ISS utility. If, on the other hand, no GBS modifications are to be made, the compressed programs can be used.

The first step in data file initialization is to copy the diskettes to hard disk. This action is accomplished by the ISS Copy/Verify utility. This step, as well as all others in data file initialization and installation of GBS, is usually performed by the software consultant.

The second step in data file initialization is to set file parameters and sizes based on the size, volume, and reporting requirements of the customer's business. Wang provides the publication, GBS Customer Survey and Configurator, to assist both the vendor and customer in obtaining these estimates. As can be seen from the excerpt of the publication (Figure 2), it presents a series of questions that assist the vendor in determining disk storage, data file, and report requirements. The answers to these questions are very useful in the data file initialization process. The boxes in Figure 2 indicate the questions that are used to calculate disk storage requirements.



**Customer Information**

A. Number of customers ..... ■

B. How many customers have:

1) 1 or 2 "ship to" addresses ..... X 1 = \_\_\_\_\_

2) 3 or 4 "ship to" addresses ..... X 2 = \_\_\_\_\_

3) 5 or 6 "ship to" addresses ..... X 3 = \_\_\_\_\_

4) More than 6 "ship to's" \_\_\_\_\_

How many total "ship to" addresses for customers with more than 6 each ..... + 2 = \_\_\_\_\_

Total "ship to sectors" (1 + 2 + 3 + 4) ..... ■

C. Customer file growth rate \_\_\_\_\_

D. Are certain customers assigned different prices than others? \_\_\_\_\_

If yes, how many customer pricing classes do you require? \_\_\_\_\_

**Inventory Information**

A. Number of products ..... ■

B. Product file growth rate \_\_\_\_\_

C. Do you have a product numbering system? \_\_\_\_\_ If so, how many characters in the product code? \_\_\_\_\_

D. How many prices are required for each product? \_\_\_\_\_

E. Are there quantity price breaks? \_\_\_\_\_

F. Do you have more than one warehouse? \_\_\_\_\_

**Invoicing**

A. Number of invoices per day: Average \_\_\_\_\_ Peak \_\_\_\_\_

B. Number of lines per invoice: Average \_\_\_\_\_ Peak \_\_\_\_\_

C. Attach a blank invoice and a completed copy of the invoice form.

D. Can present invoice format be changed? \_\_\_\_\_

E. How many invoice copies are needed? \_\_\_\_\_

F. Are there any special tax requirements (reports, etc.) \_\_\_\_\_

Explain \_\_\_\_\_

\_\_\_\_\_

**Order Entry**

A. How many open orders (orders received but not shipped) on file at peak period? ..... ■

B. Number of lines per order: Average \_\_\_\_\_ ■ Peak .....

C. Number of orders per day: Average \_\_\_\_\_ Peak .....

D. Average turnaround time of orders (in days): \_\_\_\_\_

E. Attach a blank order and a completed copy of the order form.

F. How many order copies are needed? \_\_\_\_\_

G. Can present order format be changed? \_\_\_\_\_

H. Number of orders confirmed (i.e., shipped) per day:

Average \_\_\_\_\_ Peak \_\_\_\_\_

I. Are automatic back orders a requirement? \_\_\_\_\_

Figure 2. GBS Survey on Customer, Inventory, Invoicing, and Order Entry Information

A set of programs guides the vendor through data file initialization. These programs ask the vendor to specify which applications are being selected, and after the selections have been made, a series of questions or prompts relating to the files in that system are listed. These questions on such topics as numbers of customers, products, invoices open items, and orders, are based on questions asked in the GBS Customer Survey and Configurator. An example of a data file initialization prompt is presented in Figure 3.

```
FILE INITIALIZATION PROGRAM
ENTER NO. OF FILE TO BE INITIALIZED

'X' means file not requested, '*' means file already initialized

0 - END OF KFAM FILE INITIALIZATION
1 - Customer master
2 - Inventory master
3 - A/R open item (KFAM)
4 - Open order
5 - Salesman master
6 - Invoice trans. A
7 - Invoice trans. B
8 - P.O. Activity
9 - Vendor master
10 - A/P Open item
11 - Chart of accounts
12 - Employee master
13 - Bank address
14 - Profile
15 - Product Structure
16 - Where used
```

Figure 3. An Example of a Prompt in GBS Data Initialization

On the basis of the response to these questions, the system initializes the files for the applications chosen. Table 2 presents a list of data files for each application. At this time do not be concerned with the definition of each of these files; they will be defined later in the module.

Table 2. GBS Data Files

<p><b>Invoicing/ *</b> <b>Accounts Receivable</b></p> <p>Customer File Salesman File A/R Open Item File Control File Inventory File</p>
<p><b>Order Entry/ *</b> <b>Inventory Control</b></p> <p>Inventory File Open Order File P.O. Activity File Lost Sales/Estimated Shortage File Shipping Shortage File</p>
<p><b>Accounts Payable/</b> <b>General Ledger</b></p> <p>Vendor Master File A/P Open Item File Check File Control File Chart of Accounts File Journal Entry File</p>
<p><b>Payroll</b></p> <p>Employee Master File Control File Bank Address File</p>
<p><b>Bill/ *</b> <b>Materials</b></p> <p>Product Structure File Where Used File Inventory File</p>

The asterisk (\*) in Table 2 deserves special mention. As you remember from the "Introduction to 2200 Software Course," several GBS applications are necessary prerequisites to others, that is, one cannot exist without the other. The three applications with the asterisk are part of a hierarchy. This hierarchy is depicted in Figure 4.

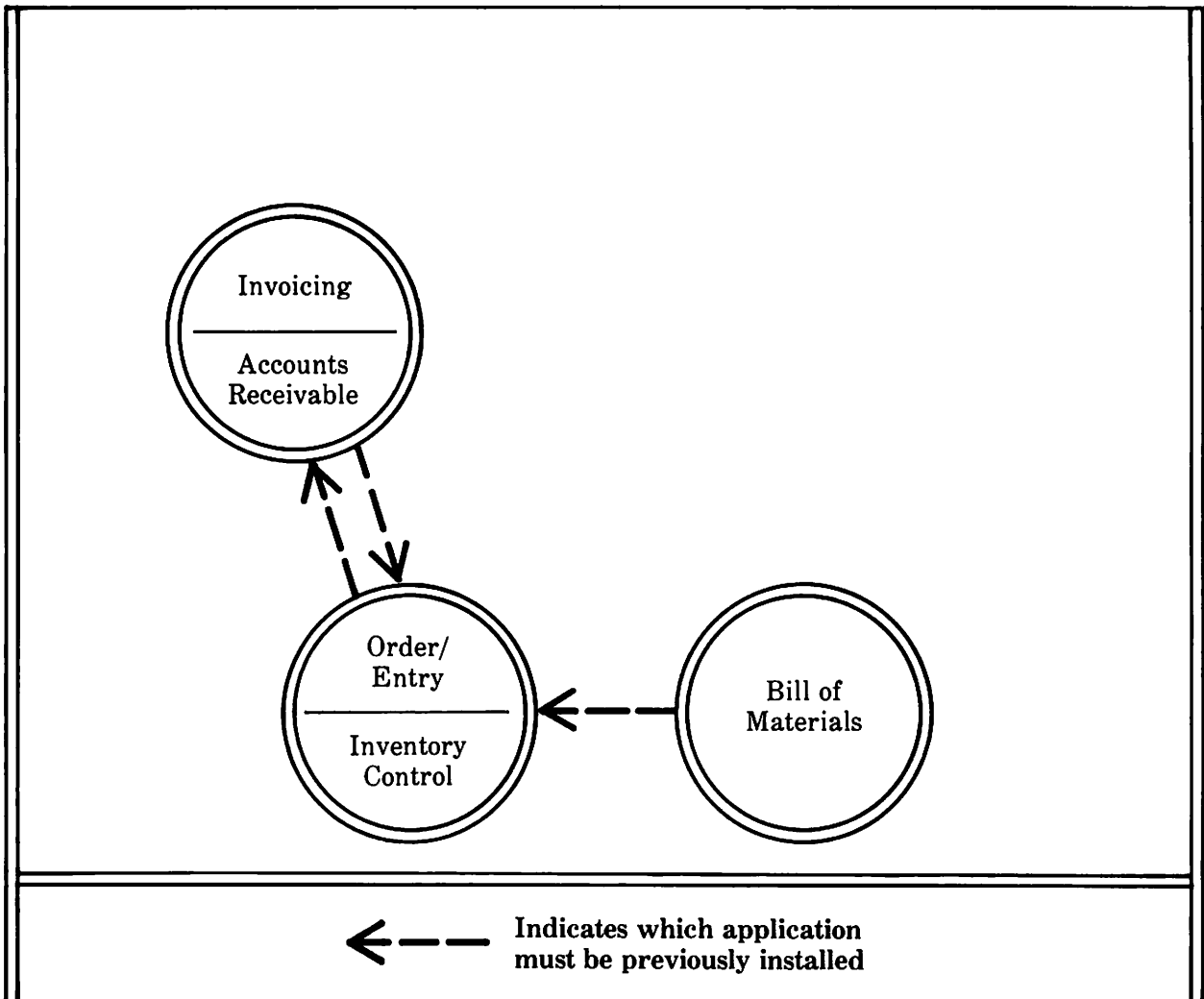


Figure 4. Dependent GBS Applications

These dependent applications share data files. Bill of Materials, for example, requires an inventory file and is therefore dependent on the GBS Order Entry/Inventory Control application. The Invoicing/Accounts Receivable and Order Entry/Inventory Control applications are packaged together and cannot be purchased separately, even though, GBS documentation lists them as separate applications.

These, then, are the basic data file initialization procedures. They are simple, user-oriented procedures which can be completed quickly and efficiently. Up to this point, our discussion of GBS has focused on the general purpose of each application, general features and benefits, minimum hardware requirements and data file initialization procedures. Now it is time to focus on

# GENERAL BUSINESS SYSTEM

the five applications. Each of the applications is discussed in terms of: purpose and features; system files; system menu and reports; and examples of relevant reports and screens. Figure 5 presents the five applications.

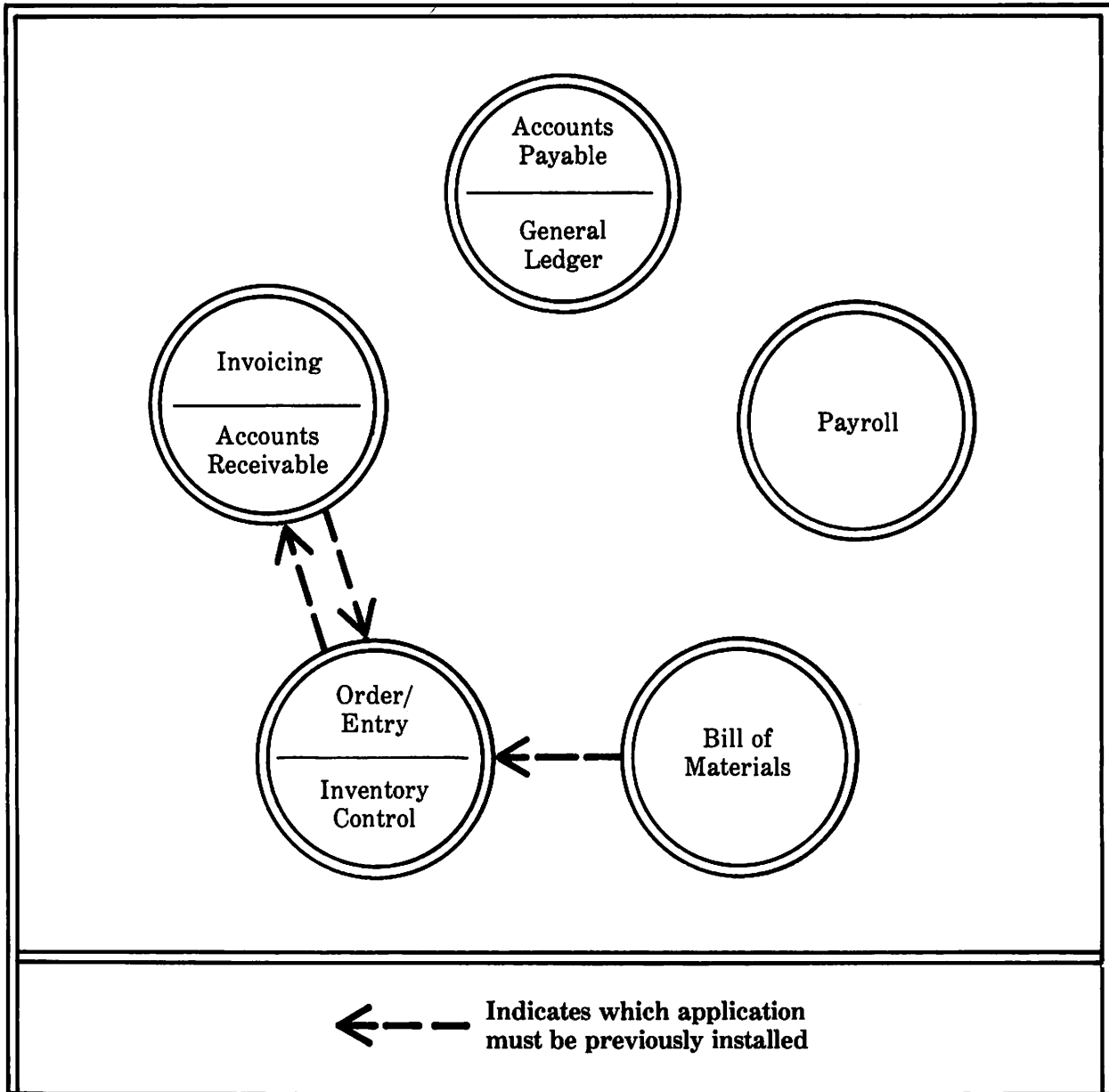


Figure 5. Five GBS Release 2.0 Applications

**INVOICING/ACCOUNTS RECEIVABLE GBS APPLICATION**

Invoice programs create invoices containing all information relevant to the customer and the salesman. They draw on the inventory file for key information including cost, description, and availability. Invoices can be printed separately or all can be printed at the completion of input. Both debit and credit invoices and debit and credit memos can be generated.

Accounts receivable programs interface with the invoice system for both open item and balance forward customers. Open item accounts show each invoice that has a balance and a total of all invoices. Balance forward accounts show a total amount due for the account. These are automatically posted when the invoicing is completed. Customer file maintenance provides for adding, editing, and deleting accounts. As invoices are paid in full, they are cleared from the account. An aged analysis can be generated which will assist in an examination of cash flow. It shows which accounts are slower in paying and which are overdue.

This application also provides Sales Analysis reports which cover sales, costs, and profit margins for the month to date and year to date. The output can be printed by customer, inventory item, or salesman. The software vendor can make additions to Sales Analysis programs which offer the customer a wide scope of analysis and reports based on sales.

**Application Data Files**

Invoicing/Accounts Receivable uses the following data files: customer, salesman, A/R open item, inventory, and control. Table 3 lists the individual fields in the customer and salesman master files. The A/R open item file, control file, and inventory file are discussed further.

- The A/R open item file lists those orders which are still outstanding. This file handles all open item or balance forward customers by carrying either a balance forward or open item record.
- The control file contains the latest invoice number, latest non-regular invoice number, latest order number and most recent accounts receivable balance. The control file is automatically updated when actions such as an invoice being created or an order being placed occur.
- The inventory file contains information on units on hand, unit prices, units sold, and products on order. The specific fields in the inventory file are presented in the order entry/inventory control application.

**GENERAL BUSINESS SYSTEM**

---

**Table 3. Fields in GBS Customer and Salesman Master File**  
 (Note: A definition of each field can be obtained  
 in The GBS/MVP System Manual)

GBS Customer Master File	GBS Salesman Master File
Customer ID Customer Name Address 1 and Address 2 City, State Zip Code Location Code Discount Class Credit Limit Salesman ID Service Charge Code Telephone Number Price Code A/R Code Partial Ship Code Current Month Cost YTD Cost Current Month Sales YTD Sales Date of Last Payment Date of Last Activity Order Total Ship to Address 1 and 2 Ship to City/State Ship to Zip Code Ship to Location Code	Salesman Number Salesman Name Territory Current Month Sales YTD Sales Current Month Cost YTD Cost

Application Menu and Examples

GBS/MVP INVOICING & ACCOUNTS RECEIVABLE SYSTEM  
Release 2.0

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	ENTER INVOICES	(09)	MAINTAIN CUSTOMER FILE
(01)	PRINT INVOICE REGISTER	(10)	MAINTAIN SALESMAN FILE
(02)	PRINT INVOICES	(11)	ENTER CASH RECEIPTS
(03)	POST INVOICES	(12)	DISPLAY/PRINT CUSTOMERS
(04)	AGE A/R & CALCULATE SERVICE CHARGES	(13)	DISPLAY/PRINT SALESMAN
(05)	PRINT A/R STATEMENTS	(14)	DISPLAY/PRINT A/R OPEN ITEMS
(06)	PRINT AGED TRIAL BALANCE	(15)	PRINT CREDIT REPORT
(07)	PURGE OPEN ITEM FILE	(16)	DISPLAY/PRINT CONTROL FILE
(08)	CLEAR FILES	(31)	SYSTEM MENU

Figure 6. Invoicing and Accounts Receivable Menu

There are several categories of activities within this menu. The first is the Invoice programs which account for the first four menu choices. The next category is the Accounts Receivable month-end procedure which include choices 04 through 07. These programs must be run sequentially to perform the A/R month-end procedure. The third category is the two programs which maintain the customer and salesman master file. These are widely used files and need frequent updating. The fourth general category is the display/print programs (menu choices 12, 13, 14, and 16) which allow the operator to view the information in crucial GBS data files. There are also other programs such as Clear Files, Enter Cash Receipts, and Print Credit Report that perform important tasks but can not be readily categorized.

An example of an Invoice and an aged trial balance report is included in Figures 7 and 8.



GENERAL BUSINESS SYSTEM.

SOLD TO:				SHIP TO:			
AUTUMN SALES				SAME			
88 SALEM STREET				(NAME AS SOLD TO UNLESS INDICATED)			
CAMBRIDGE MASS. 02121							
ORDER 000001-00 DATE 08/01/80							
CUSTOMER P.O.	CUSTOMER ID.	TERMS	SHIP VIA	P/C	SALESMAN NO.	INVOICE DATE	INVOICE NO.
PO 00001	00100	2/15 NET 30	TRUCK	P	100	08/06/80	00000001
PART NUMBER	QUANTITY	DESCRIPTION	UNIT PRICE	DISC AMOUNT	NET AMOUNT		
1000	3	25" COLOR TV (COMPLETE)	500.00		1,500.00		
2000	4	25" B/W TV (COMPLETE) SHIP WITH STAND	490.00		1,960.00		
SALES TAX 1	SALES TAX 2	SALES TAX 3	FREIGHT	INVOICE DISC	SPEC CHARGE CREDIT	INVOICE TOTAL	
62.28	93.42	124.56	145.00	345.00	23.00	3,562.26	
CASH DISCOUNT OF 2.00% IF PAID BY 08/31/80							

Figure 7. Example of a Printed Invoice (Menu Choice 02)

FUTURA TV PRODUCTS		AGED TRIAL BALANCE					09/30/80 PAGE 1		
CUST ID	CUSTOMER NAME	TELEPHONE NO	SERVICE CHARGE	CURRENT	31-60	61-90	91-OVER	TOTAL DUE	
00100	AUTUMN SALFS	617-862-1234	1,185.58	1,185.58	84,826.25	3,864.00	0.00	89,875.83	
00200	SALTY CORPORATION	617-725-4500	19.62	19.62	0.00	1,762.09	0.00	1,781.66	
00300	MARKM DEPT STORES	617-272-5000	6.18	6.18	0.00	417.50	0.00	423.68	
00400	WALDEN DEPT STORES	617-272-9550	0.00	450.00-	0.00	0.00	0.00	450.00-	
00500	DAVIDSON'S EMPORIUM	617-933-0022	29.31	29.31	0.00	2,731.22	0.00	2,760.53	
00600	WOOL-PEMNY, INC	919-774-1010	50.49	2,342.19	4,849.40	0.00	0.00	7,191.59	
00800	AJAX APPLIANCE, INC	617-227-3000	103.24	103.24	10,124.28	0.00	0.00	10,227.52	
			FINAL TOTAL	1,194.42	99,799.93	8,774.76	0.00	111,810.81	

END OF REPORT

Figure 8. Aged Trial Balance (Menu Choice 06)

## GENERAL BUSINESS SYSTEM

---

These, then, are the menu selectable choices for the GBS Invoicing/Accounts Receivable application. This application must be installed with the GBS Order Entry/Inventory Control application. These programs use five main data files: customer, salesman, A/R open item, inventory and control. Five reports and four display/print options are contained within this application. Table 4 summarizes the key characteristics of the Invoicing/Accounts Receivable application.

Table 4. Invoicing/Accounts Receivable Data Files, Reports and Display/Print Options

Data Files	Reports	Display/Print Options
Customer Salesman A/R Open Item Control Inventory	Invoice Register Invoices A/R Statements Aged Trial Balance Credit Report	Customer Salesman A/R Open Items Control File

**ORDER ENTRY/INVENTORY CONTROL: GBS APPLICATION**

Order entry programs process customer orders, update open order files, adjust and confirm orders, and print shipping papers. The invoices can be printed as desired, either at the time of entering the order, or later if delivery is delayed.

The Inventory Control programs update the inventory file with receipts and withdrawals other than those processed during invoicing. Many different types of reports are available including stock status, low stock items, inactive items information, physical inventory sheets, and recommended purchase order reports. All transactions effecting inventory are interfaced, creating audit trails and relevant reports. These reports improve the customer's cash flow by clearly defining those items in the inventory that are moving well, and those which represent a loss.

**Application Data Files**

Five main data files are used by the Order Entry/Inventory Control application.

- Open Order File contains customer ship-to and bill-to information and order date, order number, and line item information.
- Purchase Order Activity File contains a listing of active vendors and products.
- Lost Sales/Estimated Shortage File has information (e.g., order number, quantity ordered, quantity released, estimated shortage, and unit price) relating to revenue that is lost due to backordered products.
- Shipping Shortage File information stores on products where the quantity shipped is less than the quantity ordered.
- Inventory File is a very important file which is used by several applications (Invoicing/Accounts Receivable and Bill of Materials). It contains all product information, such as balances, price codes, and cost descriptions. A list of the fields in the Inventory File follows.

## GENERAL BUSINESS SYSTEM

---

Product ID  
Product prefix  
Product description  
Unit of measure  
Last activity date  
Vendor ID  
Alternate ID  
Stock location code  
On hand quantity  
On order quantity  
Allocated quantity  
Minimum (to have on-hand)  
Maximum (to have on-hand)  
Average unit cost  
Last unit cost  
Units sold current month  
Units sold YTD  
Costs of units sold current month  
Cost of units sold YTD  
Sales current month  
Sales YTD  
Unit price 1  
Unit price 2  
Unit price 3  
Unit price 4  
Unit price 5  
Weight  
Lead time  
Physical on-hand  
Physical count  
Physical counting cycle  
Lost sales current month  
Lost sales YTD  
Bill of materials flag  
Pack size  
Item minimum order  
Price break quantities

### Application Menu and Examples

In the GBS System menu there are separate menus for the Inventory and Order Entry parts of this application.

The Inventory menu is displayed in Figure 9. The first two menu choices deal with updating information on the inventory master file. The MAINTAIN INVENTORY FILE option allows the operator to add, delete, or change information in the file while the Inventory Transaction program updates the file with receipts or withdrawals. A third way to update the Inventory file is through Invoicing actions taken in the Invoicing/Accounts receivable application (remember the Order Entry/Inventory Control application is dependent on the previous installation of the Invoicing/Accounts Receivable application).

GBS INVENTORY SYSTEM  
Release 2.0

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	MAINTAIN INVENTORY FILE	(05)	PRINT PHYSICAL INVENTORY SHEETS
(01)	ENTER INVENTORY TRANSACTIONS	(06)	ENTER PHYSICAL COUNT
(02)	PRINT STOCK STATUS REPORT	(07)	PRINT PHYSICAL INVENTORY REPORT
(03)	PRINT INACTIVE/LOW STOCK REPORT	(08)	RECOMMEND PURCHASE ORDERS
(04)	DISPLAY/PRINT INVENTORY	(31)	SYSTEM MENU

Figure 9. Inventory Menu

Menu choices SFK (02), (03), (05), (07), and (08) are inventory reports that can be generated by this application. The DISPLAY/PRINT INVENTORY option enables the operator to view the information in the Inventory Master File. Figure 10 is an example of a Stock Status Report (SFK 02).

FULL REPORT OPTION		FUTURA TV PRODUCTS		STOCK STATUS REPORT						08/01/80	PAGE 1
PRODUCT ID	DESCRIPTION	LAST ACTIVITY	ON HAND	ALLOCATED	NET AVAIL	MINIMUM	ON ORDER	BACK-ORDERED	UNIT COST	TOTAL COST	
100-10	TUPE 25" COLOR	08/01/80	199	0	199	50	0	0	85.00	16,915.00	
100-20	DISPLAY GPN. 25" COLOR	08/01/80	200	0	200	50	0	0	35.00	7,000.00	
100-30	MOUNTING BKT. 25" COLOR	08/01/80	400	0	400	50	0	0	5.00	2,000.00	
1000	25" COLOR TV (COMPLETE)	08/02/80	167	2	165	50	0	0	249.10	41,599.70	
1000-100	TUBE ASSY 25" COLOR TV	08/01/80	400	12	388	50	0	2	140.00	56,000.00	
1000-110	CABINET 25" (FITS ALL)	08/01/80	594	5	589	50	0	0	34.10	20,255.40	
110-10	LEGS - 25" COLOR TV	08/01/80	890	10	880	400	0	0	2.50	2,225.00	
110-20	FRONT PANEL 25" COLOR TV	08/01/80	201	5	196	50	0	0	10.00	2,010.00	
110-30	BACK PANEL 25" COLOR TV	08/01/80	200	5	195	50	0	0	7.50	1,500.00	
110-40	SIDE PANEL 25" COLOR TV	08/01/80	410	5	405	50	0	0	5.00	2,050.00	
110-50	SCREWS - CABINET 25"	08/01/80	3,000	5	2,995	500	0	0	0.10	300.00	
2000	25" B/W TV (COMPLETE)	08/01/80	218	5	213	100	0	0	190.00	41,420.00	
2000-200	TUBE ASSY 25" B/W TV	08/01/80	400	0	400	50	0	0	80.00	32,000.00	
3000	19" COLOR TV (COMPLETE)	08/02/80	213	8	205	100	0	3	172.37	36,714.81	
3000-300	TUBE ASSY 19" COLOR TV	08/01/80	400	0	400	50	0	0	105.00	42,000.00	
3000-310	CABINET ASSY (ALL 19")	08/01/80	400	0	400	50	0	0	25.00	10,000.00	
4000	19" B/W TV (COMPLETE)	08/01/80	200	5	195	50	0	0	145.00	29,000.00	
4000-400	TUBE ASSY 19" B/W TV	08/01/80	400	0	400	50	0	0	55.00	22,000.00	
5000	12" COLOR TV (PORTABLE)	08/02/80	180	20	160	50	0	10	140.00	25,200.00	
5000-500	TUPE ASSY 12" COLOR TV	08/01/80	367	0	367	50	0	0	75.00	27,525.00	
5000-510	CABINET ASSY (ALL 12")	08/01/80	400	0	400	100	0	0	15.00	6,000.00	
6000	12" B/W TV (PORTABLE)	08/01/80	193	0	193	50	0	0	95.00	18,335.00	
6000-600	TUPE ASSY 12" B/W	08/01/80	400	0	400	50	0	0	30.00	12,000.00	
7000-700	ANTENNA ASSY (FITS ALL)	08/01/80	394	0	394	100	0	0	10.00	3,940.00	
800-10	CHASSIS 25" COLOR TV	08/01/80	400	0	400	100	0	0	15.00	6,000.00	
800-20	POWER SUPPLY (ALL 25")	08/01/80	400	0	400	100	0	0	5.00	2,000.00	
800-30	MOTHER BOARD 25" COLOR	08/01/80	400	0	400	100	0	0	40.00	16,000.00	
800-40	POWER CORD (FITS ALL)	08/01/80	400	0	400	400	0	0	5.00	2,000.00	
8000-800	CHASSIS ASSY 25" COLOR	08/01/80	398	0	398	100	0	0	65.00	25,870.00	
9000-1	25" COLOR TV MANUAL	08/01/80	521	120	401	10	100	0	22.50	11,722.50	
9000-2	B/W TV MANUAL	08/01/80	32	120	88-	10	595	0	18.50	592.00	
ZZZZZZZZZZZZ	WASH ACCOUNT	08/02/80	0	0	0	0	0	0	0.00		
REPORT TOTAL									522,174.41		

Figure 10. Stock Status Report

The Order Entry menu (Figure 11) contains a wide variety of reports and actions that can be taken after orders are received from customers (SFK 00). These reports serve the general purpose of completing documentation on orders that are received, shipped, backordered, or short shipped. As a result of orders being received and shipments made, programs in this application update the data files that are affected.

GBS ORDER ENTRY SYSTEM

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	ENTER ORDERS	(07)	PRINT LOST SALES/EST SHORTAGE REPORT
(01)	PRINT ORDER REGISTER	(08)	PRINT SHIPPING SHORTAGE REPORT
(02)	PRINT SHIPPING PAPERS	(09)	ADJUST OPEN ORDERS
(03)	CONFIRM SHIPMENTS	(10)	DISPLAY/PRINT OPEN ORDERS
(05)	POST INVOICES	(11)	ANALYZE OPEN ORDERS
(04)	PRINT SHIPPING REGISTER	(31)	SYSTEM MENU
(06)	PRINT BACKORDER SHIPPING PAPERS		

Figure 11. Order Entry Menu

Notice option (SFK 05), POST INVOICES; invoices can be posted both through this application and through the Invoicing/Accounts Receivable application. Figures 12 and 13 are examples of an order register (SFK 01) and shipping shortage report (SFK 08) respectively.



FUTUFA TV PRODUCTS				ORDER REGISTER DETAIL			08/01/80	PAGE 1	
ORDER/LN	CUST	NAME	PRODUCT ID	DESCRIPTION	QTY ORD	QTY REL	UNIT PRICE	NET AMT	CRED HOLD
000001	00100	AUTUMN SALES							
1			1000	25" COLOR TV (COMPLETE)	5	5	500.00	2,500.00	
2			2000	25" B/W TV (COMPLETE)	5	5	300.00	1,500.00	
				SHIP WITH STAND					
		ORDER TOTAL			10	10		4,000.00	
000002	00200	SALTY CORPORATION							
1			1000	25" COLOR TV (COMPLETE)	2	2	500.00	1,000.00	
2			2000	25" B/W TV (COMPLETE)	2	2	300.00	570.00	
3			3000	19" COLOR TV (COMPLETE)	3	3	450.00	1,350.00	
4			4000	19" B/W TV (COMPLETE)	3	3	200.00	540.00	
		ORDER TOTAL			10	10		3,460.00	
000003	00300	MAKEM DEPT STORES							
1			5000	12" COLOR TV (PORTABLE)	10	10	175.00	1,750.00	
2			6000	12" B/W TV (PORTABLE)	5	5	150.00	750.00	
		ORDER TOTAL			15	15		2,500.00	
000004	00400	WALDEN DEPT STORES							
1			1000	25" COLOR TV (COMPLETE)	2	2	879.00 **	1,740.42	
2			3000	19" COLOR TV (COMPLETE)	100	100	375.00	37,500.00	
3			5000	12" COLOR TV (PORTABLE)	2	2	175.00	350.00	
		ORDER TOTAL			104	104		39,590.42	**
000005	00500	DAVIDSON'S EMPORIUM							
1			1000	25" COLOR TV (COMPLETE)	100	80	425.00	34,000.00	
2			1000-100	TURE ASSY 25" COLOR TV	2	2	150.00	300.00	
3			1000-110	CABINET 25" (FITS ALL)	6	6	75.00	450.00	
4			7000-700	ANTENNA ASSY (FITS ALL)	12	12	25.00	270.00	
5			8000-800	CHASSIS ASSY 25" COLOR	12	12	75.00	1,026.00	
		ORDER TOTAL			132	112		36,046.00	**
000006	00600	WOOL-PENNY, INC							
1			1000	25" COLOR TV (COMPLETE)	3	3	500.00	1,500.00	
2			2000	25" B/W TV (COMPLETE)	3	3	300.00	900.00	
3			3000	19" COLOR TV (COMPLETE)	3	1	450.00	450.00	
4			4000	19" B/W TV (COMPLETE)	2	2	200.00	400.00	
5			5000	12" COLOR TV (PORTABLE)	2	2	175.00	350.00	
6			6000	12" B/W TV (PORTABLE)	20	20	150.00	2,850.00	
		ORDER TOTAL			33	31		6,450.00	**

Figure 12. Order Register

FUTURA TV PRODUCTS			SHIPPING SHORTAGE DETAIL			08/01/80	PAGE 1	
PRODUCT ID	DESCRIPTION	ORDER/LINE NO.	QTY ORD	QTY SHIPPED	QTY SHORT	UNIT PRICE	AMT SHORT	B/O FLAG
1000	25" COLOR TV (COMPLTE)	000019-00 1	10	5	5	500.00	2,500.00	
	PRODUCT TOTALS		10	5	5		2,500.00	
	REPORT TOTAL		10	5	5		2,500.00	

Figure 13. Shipping Shortage Report

**GENERAL BUSINESS SYSTEM**

---

The Order Entry/Inventory Control application encompasses a wide variety of activities related to receiving and shipping product orders and the updating of a company's inventory. This GBS application is dependent on the installation of the Invoicing/Accounts Receivable application. Table 5 shows the data files, reports, and display/print options in the Order Entry/Inventory Control application.

Table 5. Order Entry/Inventory Control Data Files, Reports, and Display/Print Options

Data Files	Reports	Display/Print Options
Open Order File P.O. Activity File Lost Sales/Estimated Shortage File Shipping Shortage File Inventory File	Stock Status Report Inactive/Low Stock Report Physical Inventory Sheets Physical Inventory Report Order Registers Shipping Papers Shipping Register Papers Backorder Shipping Lost Sales/Estimates Shortage Report Shipping Shortage Report	Inventory File Open Orders

**ACCOUNTS PAYABLE, GENERAL LEDGER GBS APPLICATION**

The Accounts Payable programs monitor a company's payments for products or services. The programs do this by posting the vendor journal, storing open items, and cataloging distribution amounts for interfacing with the general ledger. Checks can be printed for either all accounts or selected ones. General Ledger programs perform the usual routines of posting accounts (journal entries), keeping track of debits and credits, and printing financial reports. Interfacing with the journals (e.g., accounts receivable, vendor, payroll) can be accommodated when the other related GBS systems are in use. A software vendor can modify the system to provide automatic posting of depreciation. This is ordinarily a tedious and time consuming job. But programmed depreciation can guarantee accuracy while saving high and repeated labor costs.

**Application Data Files**

This application uses six data files.

- Vendor Master File contains name and address information as well YTD discounts, G/L account numbers, date of last payment, and balance due.
- A/P Open Item File contains information on all accounts that are still open or owed to vendors.
- Check File has information on vendors and checks, including gross amount discount amount, and a check code for checks to vendors.
- Control File contains the latest daily information on balances for check payments, invoices, credit memos, discounts taken, liability income, and expense accounts.
- Chart of Accounts File lists different types of accounts in terms of whether they are assets, liabilities/equity, sales/revenue, or expenses. Current and YTD debits and credits are listed as well as net activity for the previous 12 months.
- Journal Entry File contains information on the description of the journal entry, an audit trail (reference) back to the source document, and the amount of the transaction.

Application Menu and Examples

In the GBS main menu, there are separate entries for the Accounts Payable (Figure 14) and General Ledger (G/L) Menus.

GBS ACCOUNTS PAYABLE SYSTEM  
Release 2.0

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	ENTER TRANSACTIONS	(06)	MAINTAIN VENDOR FILE
(01)	PRINT CASH REQUIREMENTS	(07)	MAINTAIN OPEN ITEM FILE
(02)	SELECT ITEMS FOR PAYMENT	(08)	DISPLAY/PRINT VENDOR MASTER FILE
(03)	PRINT CHECKS	(09)	DISPLAY/PRINT OPEN ITEM FILE
(04)	PRINT CHECK REGISTER	(10)	REPORT ON DISTRIBUTION
(05)	PURGE PAID ITEMS	(31)	SYSTEM MENU

Figure 14. Accounts Payable Menu

Menu choices 00-05 are programs devoted to accounts payable posting, bill paying, and account updating. These programs must be executed sequentially. The second half of the menu (choices 06-10) allows the operator to list and update the files used during accounts payable operations. Figure 15 presents an example of Accounts Payable transaction records.

ALL OPTION		FUTURA TV PRODUCTS				APP TRANSACTION ENTRY RECORDS				08/01/80		PAGE ?
VEND ID	NAME	INVOICE NO	INVOICE DATE	GROSS AMT	DISC %	DISC AMT	NET AMT	DUE DATE	T/C	DATE PAID	CHECK NO	P.O. NO
100	SEMI-CONDUCTOR, INC	INV 0001	08/01/80	100.00	0.00	0.00	100.00	08/16/80	I			0 PO 0000.
200	MOTOR WHOLESALER, INC	INV 0002	08/04/80	1,000.00	0.00	0.00	1,000.00	09/03/80	I			0 PO 0002.
300	UNION SUPPLIES, INC	C 000001	08/01/80	19.00	0.00	0.00	19.00	08/13/80	C			0 INV 4.
300	UNION SUPPLIES, INC	INV 33	08/03/80	1,900.00	0.00	0.00	1,900.00	08/04/80	I			0 PO 9.
500	ALLIED SUPPLIES	INV 67	08/01/80	12.50	0.00	0.00	12.50	08/16/80	I			0 PO 0004.

NO. OF RECORDS PROCESSED = 5

Figure 15. Accounts Payable Transaction Records

The structure of the General Ledger menu (Figure 16) is similar to the Accounts Payable menu.

GBS GENERAL LEDGER SYSTEM  
Release 2.0

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	JOURNAL ENTRY	(07)	CLEAR FILES (End of Month)
(01)	TRIAL BALANCE REPORT	(08)	MAINTAIN CHART OF ACCOUNTS
(02)	POST J/E TO CHART OF ACCOUNTS	(09)	MAINTAIN CONTROL FILE
(03)	PRINT INCOME STATEMENT	(10)	DISPLAY/PRINT CHART OF ACCOUNTS
(04)	PRINT BALANCE SHEET	(11)	DISPLAY/PRINT CONTROL FILE
(05)	PRINT BUDGET REPORT	(12)	DISPLAY/PRINT JOURNAL ENTRY
(06)	PRINT SCHEDULE REPORT	(31)	SYSTEM MENU

Figure 16. General Ledger Menu

The programs represented by the menu choices on the left side of the G/L menu (00-06) are concerned with G/L posting, updating, and financial report writing. Menu choices 07 through 12 are devoted to maintaining and listing the files needed for General Ledger operations. Figure 17 is an example of a budget report (SFK 05).

FUTURA TV PRODUCTS		CURRENT MONTH		YEAR TO DATE		PAGE 1
FUTURA TV		BUDGET	BUDGET	BUDGET	BUDGET	
GROSS SALES		LAST YEAR	LAST YEAR	LAST YEAR	LAST YEAR	X
		ACTUAL	LAST YEAR	ACTUAL	LAST YEAR	X
GROSS SALES	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
LESS SALES RETURNS & ALLOWANCES	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
LESS SALES DISCOUNTS	\$201.60	\$0.00	\$0.00	\$201.60	\$0.00	
NET SALES	\$201.60	\$0.00	\$0.00	\$201.60	\$0.00	
OTHER REVENUE	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
INTEREST INCOME	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
OTHER INCOME	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
GROSS REVENUES	\$201.60	\$0.00	\$0.00	\$201.60	\$0.00	
COSTS AND EXPENSES						
RAW MATERIALS PURCHASES	\$43,238.40	\$0.00	\$0.00	\$43,238.40	\$0.00	
DIRECT LABOR	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
INDIRECT LABOR	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
BUILDING MAINTAINANCE & REPAIR	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
MISC. FACTORY OVERHEAD	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
SALES/SALARY OVERHEAD	\$0.00	\$0.00	\$0.00	\$0.00	\$0.00	
ADVERTISING EXPENSE	\$1,000.00	\$0.00	\$0.00	\$1,000.00	\$0.00	

Figure 17. Budget Report



FUTURA TV PRODUCTS		RUDGET REPORT AS OF 08/01/80		PAGE 2	
FUTURA TV		CURRENT MONTH		YEAR TO DATE	
	ACTUAL	BUDGET	LAST YEAR	ACTUAL	'BUDGET
		LAST YEAR		LAST YEAR	LAST YEAR
			X		X
			X		X
DELIVERY EXPENSE	\$0.00	\$0.00		\$0.00	\$0.00
OFFICER'S SALARIES	\$0.00	\$0.00		\$0.00	\$0.00
MISC. GENERAL EXPENSE	\$0.00	\$0.00		\$0.00	\$0.00
INTEREST EXPENSE	\$0.00	\$0.00		\$0.00	\$0.00
FACTORY SUPPLIES EXPENSE	\$0.00	\$0.00		\$0.00	\$0.00
OFFICE SUPPLIES	\$0.00	\$0.00		\$0.00	\$0.00
DEPRECIATION EXPENSE	\$0.00	\$0.00		\$0.00	\$0.00
DEPRECIATION EXPENSE (BUILDING)	\$0.00	\$0.00		\$0.00	\$0.00
DEPRECIATION EXPENSE (M & E)	\$0.00	\$0.00		\$0.00	\$0.00
DEPRECIATION EXPENSE (F & F)	\$0.00	\$0.00		\$0.00	\$0.00
DEPRECIATION EXP (DEL EQUIP)	\$0.00	\$0.00		\$0.00	\$0.00
INCOME TAX	\$0.00	\$0.00		\$0.00	\$0.00
INSURANCE EXPENSE	\$0.00	\$0.00		\$0.00	\$0.00
TOTAL EXPENSES	\$44,238.40	\$0.00		\$44,238.40	\$0.00
NET INCOME	\$44,440.00	\$0.00		\$44,440.00	\$0.00

Figure 17. Budget Report (Continued)

**GENERAL BUSINESS SYSTEM**

---

The Accounts Payable/General Ledger application is an independent application; it does not require the previous installation of another GBS module. The capabilities of General Ledger are, however, enhanced by the Invoicing/Accounts Receivable application. Table 6 presents a summary of the key characteristics of the Accounts Payable/General Ledger application.

**Table 6. Accounts Payable/General Ledger Data Files, Reports, and Display/Print Options**

<b>Data Files</b>	<b>Reports</b>	<b>Display/Print Options</b>
Vendor Master File A/P Open Item File Check File Control File Chart of Accounts File Journal Entry File	Cash Requirements Checks Check Register Distribution Report Trial Balance Income Statement Balance Sheet Budget Report Schedule Report	Vendor Master File Open Item File Chart of Accounts Control File Journal Entry

**PAYROLL GBS APPLICATION**

The Payroll programs calculate earnings, taxes, and deductions from entered data or previously stored information for hourly, salaried, and other employees. Most employers have some payroll accounts of each type. Payment can be by check, cash, and direct bank deposit. This provides the widest possible base for a payroll system.

There are probably more governmental reports required by the payroll than any other accounting function in a business. The employee master file is comprised of six parts that accommodate a wide scope of payments and federal, state, and local taxes. A different number of dependents can be used to calculate deductions for each type of tax. Sometimes employees overstate or understate their deductions so that the tax deduction amount will better suit their individual needs. The payroll programs provide for all these choices. Deductions for many things, including union dues, charitable contributions, and savings bonds, are available. There is also space for options such as pension plans, earnings and deduction codes, and bank account numbers (for direct deposit).

The control file contains company data, FICA (social security) payroll limits, and FICA percentages. Because the FICA parameters are frequently changed by the Federal government, much time and money could be spent in modifying the payroll

files to calculate the proper amounts. The GBS payroll programs change these factors with a minimum of time and effort.

Application Data Files

The payroll application uses three data files.

- Employee Master File contains information on each employee including personal information, earnings, tax and deductions.
- Bank Address File lists the address of the bank to which direct deposits are sent.
- Control File contains company information including data on earning descriptions, tax descriptions, deduction descriptions, and tax switches.

Application Menu and Examples

The payroll cycle is the complete payroll process for one pay period. The Payroll menu (Figure 18) is arranged so that menu choices 00-07 perform operations for a complete pay cycle when executed in sequence.

GBS/MVP PAYROLL SYSTEM  
Release 2.0

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	START PAY CYCLE	(08)	EMPLOYEE MASTER FILE MAINTENANCE
(01)	ENTER HOURS & DEDUCTIONS	(09)	BANK ADDRESS FILE MAINTENANCE
(02)	CALCULATE GROSS AND NET PAY	(10)	CONTROL FILE MAINTENANCE
(03)	ENTER ADJUSTMENTS	(11)	EMPLOYEE MASTER FILE INQUIRY/LIST
(04)	PRINT EMPLOYEE REGISTERS	(12)	BANK ADDRESS FILE INQUIRY LIST
(05)	PRINT CHECKS, MEMOS, 941s, W-2s	(13)	CONTROL FILE/INQUIRY/LIST
(06)	PRINT RECONCILIATION LIST	(31)	SYSTEM MENU
(07)	END OF PAY CYCLE/PERIOD		

Figure 18. Payroll Menu

Menu choices 08-13 serve to primarily list and maintain the files used in the payroll cycle. Figure 19 is an example of an employee register.

EMPLOYEE NAME	E A R N I N G S		T A X E S		D E D U C T I O N S		CURRENT DATE 08/01/80	PAGE 1
	CURRENT QTR TO DATE	YR TO DATE	CURRENT QTR TO DATE	YR TO DATE	CURRENT QTR TO DATE	YR TO DATE		
M H HARRISON	260.00	260.00	32.27	32.27	2.60	2.60		
CHARL'S PEKING	0.00	0.00	0.00	0.00	0.00	0.00		
SHARON A. FALLON	0.00	0.00	0.00	0.00	0.00	0.00		
JOHN KELLY	0.00	0.00	0.00	0.00	0.00	0.00		
ALAN P. GOLDMAN	0.00	0.00	0.00	0.00	0.00	0.00		
THOMAS J. MURPHY	0.00	0.00	0.00	0.00	0.00	0.00		
HAROLD S. EVANS	0.00	0.00	0.00	0.00	0.00	0.00		
STEPHEN J. HINIS	0.00	0.00	0.00	0.00	0.00	0.00		

DEPARTMENT - DETAIL REGISTER  
 FUTURA TV PRODUCTS  
 CHECK DATE 08/31/80  
 PERIOD ENDING 08/25/80

Figure 19. Employee Register

The Payroll application, then, provides programs to complete the entire pay cycle. It is a comprehensive package designed to meet all the payroll needs of small-to medium-sized businesses. Table 7 presents a summary chart listing payroll data files, reports, and inquiry/list options.

Table 7. Payroll Data Files, Reports, and Inquiry/List Options

Data Files	Reports	Inquiry/List Options
Employee Master File Bank Address File Control File	Employee Register Checks, Memos 941A and W-2 Forms Reconciliation Lists	Employee Master File Bank Address File Control File

**BILL OF MATERIALS GBS APPLICATION**

The Bill of Materials (BOM) application provides a variety of manufacturing related functions. This application is not designed for customers in retail sales or distributors of finished products. It requires a previous installation of the Order Entry/Inventory Control GBS application. The primary purpose of BOM is to break a product down into its various component parts and their prices. The customer, then, will be able to analyze the total cost in terms of a number of component costs.

In a manufacturing environment, the BOM application greatly assists the customer in analyzing complex business decisions based on such factors as varying cost and availability of different component parts. When, for example, the customer is wondering about the impact of doubling production on a certain product, the BOM application can present reports on the results of this possible change in terms of its impact on available inventory and cost.

**Application Data Files**

The Bill of Materials application uses three data files.

- Product Structure File contains records consisting of assembly information combined with information on the use of each assembly product. In essence, this file has a list of components for each assembly in the manufacturing system.

**GENERAL BUSINESS SYSTEM**

---

- Where Used File contains information on component parts and their relationship to the assembly products in the Product Structure file.
- Inventory File - Each product in the Product Structure file must have a corresponding record in the Inventory file (in GBS, the Order Entry/Inventory Control application is where the Inventory file is maintained and used). The BOM Parts Explosion, Gross Requirements Explosion, and Costing from Parts programs use the Inventory file.

**Application Menu and Examples**

The Bill of Materials menu (Figure 20) may look familiar because it was presented in the discussion of Inventory Control. The Bill of Materials application begins with option 09 and proceeds through 16, or in other words, all the programs in the right column of the menu.

**GBS/MVP INVENTORY SYSTEM**  
Release 2.0

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	MAINTAIN INVENTORY FILE	(09)	MAINTAIN BILL OF MATERIALS FILES
(01)	ENTER INVENTORY TRANSACTIONS	(10)	ASSIGN LOW LEVEL CODES
(02)	PRINT STOCK STATUS REPORT	(11)	B.O.M. PARTS EXPLOSION
(03)	PRINT INACTIVE/LOW STOCK REPORT	(12)	DISPLAY/PRINT WHERE USED FILE
(04)	DISPLAY/PRINT INVENTORY	(13)	DISPLAY/PRINT MULTI-LEVEL IMPLOSION
(05)	PRINT PHYSICAL INVENTORY SHEETS	(14)	DISPLAY/PRINT SINGLE LEVEL EXPLOSION
(06)	PRINT PHYSICAL INVENTORY REPORT	(15)	GROSS REQUIREMENTS EXPLOSION
(07)	ENTER PHYSICAL COUNT	(16)	COSTING FROM PARTS
(08)	RECOMMEND PURCHASE ORDERS	(31)	SYSTEM MENU

Figure 20. Bill of Materials Menu

There are three key terms that further help to explain the operation of these programs: level codes, explosion, and implosion. A level code (from 0-31) is assigned to a finished product and all its sub-components in the hierarchy of production; level 0 is a finished product not used for any higher assembly product. An explosion takes a finished product and breaks it down into all its various components (level 1, 2, 3 and so on). An implosion is exactly the opposite. It takes the components and searches for higher level products in which it is used. Menu choices 10-16 are divided between explosion and implosion reports. Choices 12 and 13 (DISPLAY/PRINT WHERE USED FILE and DISPLAY/PRINT MULTI-LEVEL IMPLOSION) are implosions and the remaining options are explosions. Figure 21 is an example of a gross requirements explosion.

FUTURA TV PRODUCTS					
WITH NETTING		BILL OF MATERIALS GROSS REQUIREMENTS EXPLOSION			
		08/01/80 PAGE 1			
PRODUCT ID - 1000		25" COLOR TV (COMPLETE)			
ON HAND 167 ON ORDER 0		REQUESTED 1690			
LEVEL	PRODUCT ID	DESCRIPTION	ON HAND	REQUIRED	NOT AVAILABLE
1	1000-100	TUBE ASSY 25" COLOR TV	400	1523	1123
2	100-10	TUBE 25" COLOR	199	1123	924
2	100-20	DISPLAY GEN. 25" COLOR	200	1123	923
2	100-30	MOUNTING BKT. 25" COLOR	400	4492	4092
1	1000-110	CABINET 25" (FITS ALL)	594	1523	929
2	110-10	LFGS - 25" COLOR TV	890	3716	2826
2	110-20	FRONT PANFL 25" COLOR TV	201	929	728
2	110-30	BACK PANFL 25" COLOR TV	200	929	729
2	110-40	SIDE PANFL 25" COLOR TV	410	929	519
2	110-50	SCREWS - CABINET 25"	3000	14864	11864
1	7000-700	ANTENNA ASSY (FITS ALL)	394	1523	1129
1	8000-800	CHASSIS ASSY 25" COLOR	398	1523	1125
2	800-10	CHASSIS 25" COLOR TV	400	1125	725
2	800-20	POWER SUPPLY (ALL 25")	400	1125	725
2	800-30	MOTHER BOARD 25" COLOR	400	1125	725
2	800-40	POWER CORD (FITS ALL)	400	1125	725

OF 1690 REQUESTED, 167 ARE ON HAND  
NUMBER UNAVAILABLE = 1523

Figure 21. Gross Requirements Explosion

## GENERAL BUSINESS SYSTEM

---

The Bill of Materials (BOM) application is a relatively new addition to GBS. It serves to facilitate the analysis of manufacturing related functions by specifying the structural relationships among components of finished goods. Table 8 summarizes the data files, reports and display/print options in the BOM application.

Table 8. Bill of Materials Data Files, Reports, and Display/Print Options

Data Files	Reports	Display/Print Options
Product Structure File Where Used File Inventory File	BOM Parts Explosion Gross Requirements Explosion Costing from Parts	Where Used File Multi-Level Implosion Single Level Explosion

### APPLICATIONS SUMMARY

These, then, are the five GBS applications. They serve a wide variety of business and accounting needs for small-to-medium sized businesses. GBS, a system that provides many benefits to the user, has been continually improved and updated. It is an easy-to-use, flexible and powerful system that frequently results in time savings for the customer.

GBS is also an extensive system. It uses approximately 20 different data files and generates over 40 separate reports. Figure 22 depicts the five GBS applications and summarizes many of the system's files and reports.



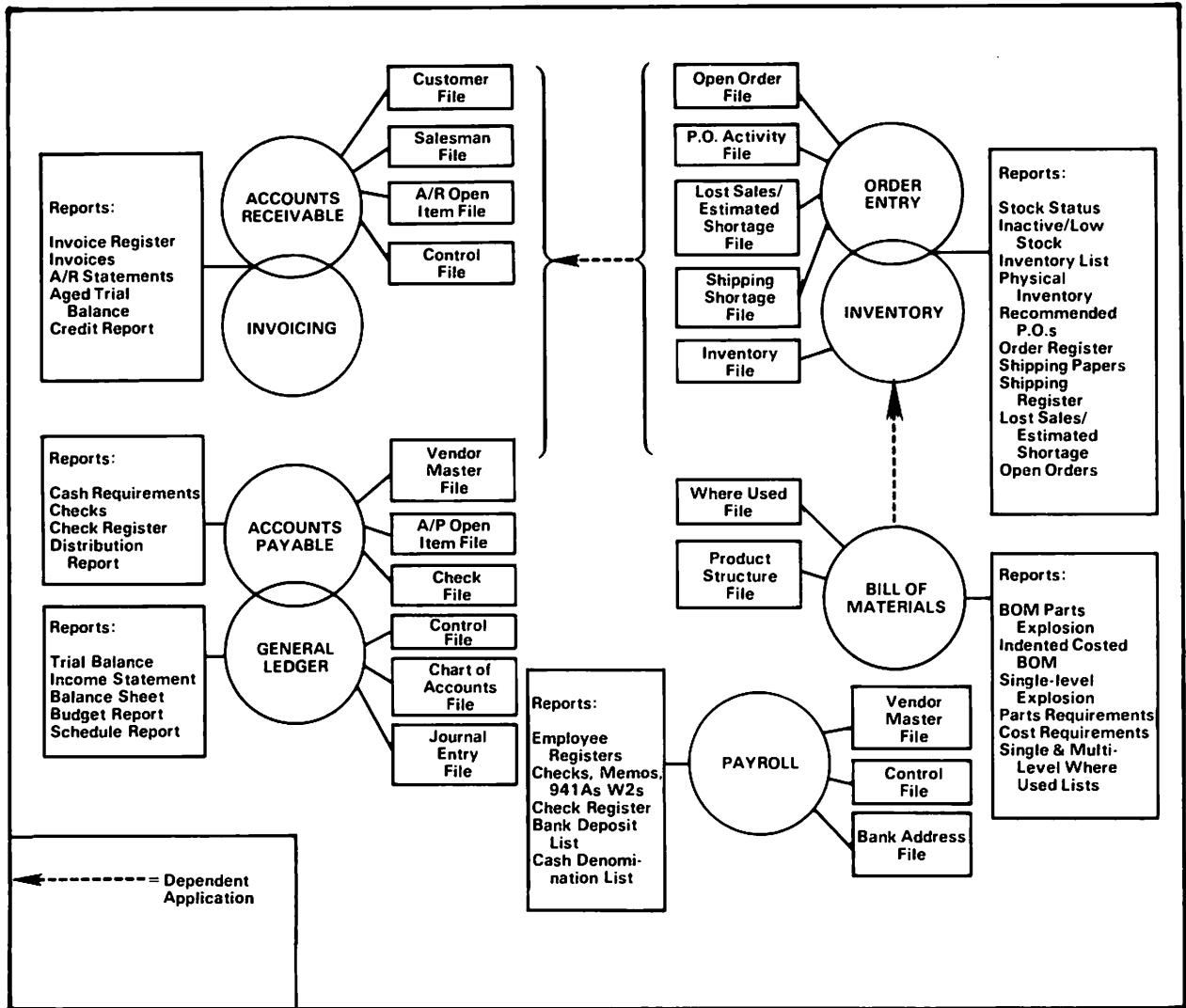


Figure 22. Summary of GBS

Before concluding this treatment of GBS, three additional topics must be discussed. Two of these topics are very important to all GBS applications. The GBS Utilities and GBS File Reorganization programs are used throughout the system. The third topic is the new Inventory Management System released by Wang; this system shares many characteristics with GBS and also serves the same small- to medium-sized business market.

GENERAL BUSINESS SYSTEM.

---

GBS UTILITIES

The GBS Utilities are accessed through the GBS main menu. Their primary purpose is to assure smooth operation of the system and data security. The GBS Utilities menu is shown in Figure 23.

GBS/MVP UTILITIES MENU  
Release 2.0

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	PRINT MAINTENANCE AUDIT REPORT	(06)	DELETE INCOMPLETE INVOICES, ORDERS
(01)	DISPLAY/PRINT TRANSACTIONS	(07)	RESET ALLOCATED (BACKORDERED) AMOUNTS
(02)	BUILD ACCOUNTS RECEIVABLE FILE	(08)	DISPLAY/PRINT CONTROL FILE
(03)	RESET ACCESS TABLES	(09)	DISPLAY KFAM QUEUE
(04)	RESTORE DATA FROM BACKUP DISK	(31)	SYSTEM MENU
(05)	COPY DATA TO BACKUP DISK		

Figure 23. GBS Utilities Menu

Notice that two utilities assist the operator in backup procedures. The Audit report and display/print options (transactions and control file) are useful in tracing the flow of information through the system and locating possible irregularities in the data base. The Build Accounts Receivable File program is run during installation to build an A/R Open Item file; this build program is also provided to facilitate conversion from an existing accounting system to GBS. Many of the remaining utilities serve to reset tables or amounts and delete incomplete transactions; these housekeeping functions are, of course, important to efficient system operation.

## GBS FILE REORGANIZATION PROGRAMS

COMPANY NAME  
GBS/MVP FILE REORGANIZATION MENU  
Release 2.0

SFK	PROGRAM NAME	SFK	PROGRAM NAME
(00)	REORGANIZE CUSTOMER MASTER FILE	(05)	REORGANIZE BILL OF MATERIALS FILES
(01)	REORGANIZE INVENTORY MASTER FILE	(06)	REORGANIZE A/P OPEN ITEM FILE
(02)	REORGANIZE SALESMAN MASTER FILE	(07)	REORGANIZE CHART OF ACCOUNTS FILE
(03)	REORGANIZE OPEN ORDER FILE	(08)	REORGANIZE VENDOR MASTER FILE
(04)	REORGANIZE P.O. ACTIVITY FILE	(31)	SYSTEM MENU

Figure 24. GBS File Reorganization Menu

All of the programs on the GBS File Reorganization menu (Figure 24) accomplish the same task. They purge deleted records from their respective files. Reorganization occurs automatically after the appropriate menu choice has been taken. These File Reorganization Programs help to assure more efficient use of file space and the smoother system functioning. The File Reorganization Menu is directly accessed from the GBS System main menu.

## INVENTORY MANAGEMENT SYSTEM

The Inventory Management System has been recently developed by Wang to also serve the small- to medium-sized business market. Its specific focus is on the retail and wholesale distribution marketplace. The Inventory Management System permits the user to convert virtually any control-oriented inventory system into a highly sophisticated management system. Because the Inventory Management System requires an inventory file, the GBS Order Entry/Inventory Control application must be previously installed.\*

---

\* The Inventory Management System can also be installed with inventory control systems from other sources but it is highly recommended that the GBS Order Entry/Inventory Control application be used.

## GENERAL BUSINESS SYSTEM

The Inventory Management System contains four major components:

1. profile maintenance program
2. demand simulator
3. reorder simulator
4. utility subroutines

These programs provide the functional capability to provide statistical sales forecasting based upon the inventory files. It uses the inventory file to create a profile which, among other things, reflects the fluctuation in seasonal sales. This is a tool that can be used to prevent overstocking of seasonal items, and help insure a closer match between supply and demand. The cash flow of the business is thereby increased because funds are not tied up in inactive inventory. The Inventory Management System, then, can assist in improving the efficiency and effectiveness of business by:

- Accurately forecasting sales demand.
- Incorporating forecasts into reordering recommendations.
- Generating a set of conditions that ensure fast and economical customer service.

With the release of the Inventory Management System, Wang again demonstrates its commitment to provide 2200 Series product line customers with a total hardware-software-service solution. GBS is, of course, a complete software package for general business applications. Inventory Management Systems is a more specialized product but it also is an important addition to 2200 Software resources.

**SUMMARY**

GBS has been the focus of this module. It is a comprehensive general business system developed by Wang, licenced to independent software consultants or vendors, and installed in many 2200 Series computer systems. It is important to remember that vendors modify and install GBS for the customer; and the main role of the analyst is to monitor vendor performance in GBS installation and to assist in answering technical questions. GBS is also an example of a predeveloped package that is usually modified or customized, to at least some extent, by the vendor. It is, therefore, a mixture of a turnkey and customized program - or what is referred to as a limited customization application. The turnkey parts and standard conventions for GBS are well documented in the individual application's User and System manuals.

These GBS training materials have covered the following topics:

- Purpose and parts of GBS.
- Features and benefits.
- Minimum hardware configuration.
- Data file initialization and the GBS configurator.
- Overall structure of GBS (dependent and independent operations).
- Description of each of the five applications in terms of purpose, data files, and application menu with examples of screens or reports.
- GBS utilities and file reorganization programs.
- Inventory Management System.

When working on GBS projects, you will have to investigate the applications in greater detail than was presented in these materials. The best sources for you are the Technical, User, and System manuals for each application and the program code itself. Because of the size and complexity of GBS, there is not a central users manual such as those for ISS or IDEAS. Instead, there are separate manuals for each application (with the exception of a single user manual for the combined Order Entry/ Invoicing and Accounts Receivable/Inventory Control applications). The purpose of these training materials has been to provide you with an understanding of the features, benefits, and capabilities of the entire GBS package and to provide a firm basis for further, more in-depth, technical analysis of GBS.

REVIEW QUESTIONS

1. Which of the following statements is true?
  - a. GBS is a turnkey program that is licenced directly to customers.
  - b. Vendors purchase GBS from Wang, make requested modifications, and install it in the customer's 2200 system.
  - c. GBS is a joint Wang-vendor development project; Wang has developed most of the applications but vendors have contributed the new Bill of Materials and Inventory Control systems.
  - d. GBS is a custom built program that is designed to meet the specific data processing needs of the customer.
  
2. Which of the following is an associated system of GBS that shares many of its characteristics and focus on the small-to medium-sized business market place?
  - a. Payroll
  - b. Inventory Management
  - c. IDEAS
  - d. General Ledger
  
3. Which of the following GBS features allows service bureaus and other businesses to use the same GBS applications with two or more companies?
  - a. Service Bureau Utilities
  - b. Super Simultaneous Selection System (SSSS)
  - c. Multicompany Capabilities
  - d. Password and File Back-up Capability
  
4. Which of the following applications is new with GBS Release 2.0?
  - a. Payroll
  - b. Bill of Materials
  - c. Inventory Management
  - d. Accounts Payable/General Ledger
  
5. GBS Release 2.0 would NOT work with which of the following?
  - a. 2200 MVP with 64K
  - b. 2236 DE Interactive Terminal
  - c. 2270A Diskette Drive
  - d. Any 132 Character Per Line Printer

## GENERAL BUSINESS SYSTEM

---

6. What publication can be used by the vendor to establish the customers' file sizes and report requirements?
  - a. GBS Customer Survey and Configurator
  - b. GBS Technical and User Manual
  - c. GBS Sample File Size and Report Manual
  - d. Computerworld
  
7. The customer and salesman master files are primarily used in which application?
  - a. Payroll
  - b. Bill of Materials
  - c. Invoicing/Accounts Receivable
  - d. Accounts Payable/General Ledger
  
8. The Employee Data File is used in which application?
  - a. Payroll
  - b. Bill of Materials
  - c. Invoicing/Accounts Receivable
  - d. Accounts Payable/General Ledger
  
9. If a customer wanted a trial balance income statement and budget report, which application should be recommended?
  - a. Invoicing/Accounts Receivable
  - b. Order Entry/Inventory Control
  - c. Accounts Payable/General Ledger
  - d. Payroll
  
10. If a customer wanted a shipping register and a lost sales/estimated shortage report, which application should be recommended?
  - a. Invoicing/Accounts Receivable
  - b. Order Entry/Inventory Control
  - c. Accounts Payable/General Ledger
  - d. Payroll
  
11. Which of the following actions should be taken to purge the data files of deleted records?
  - a. Invoke GBS Utilities
  - b. File reorganization
  - c. Use the special Stalin-Purge program
  - d. Use File Maintenance option in each application menu

12. Which system should be used to gain sales forecasts and to assist in reordering decisions?
- a. GBS
  - b. Order/Entry/Inventory Control
  - c. Inventory Management System
  - d. Invoicing, Accounts Receivable



ANSWERS

1. b
2. b
3. c
4. b
5. c
6. a
7. c
8. a
9. c
10. b
11. b
12. c

**MODULE 3**

**INQUIRY DATA ENTRY ACCESS SYSTEM (IDEAS)**

CONTENTS

ABSTRACT.....TAS-1

OBJECTIVES.....TAS-2

MATERIALS REQUIRED.....TAS-2

DIRECTIONS FOR COMPLETION.....TAS-2

INTRODUCTION.....TAS-3

FEATURES AND BENEFITS.....TAS-7

HARDWARE REQUIREMENTS.....TAS-9

DATA FILE UTILITIES.....TAS-13

    Existing Data File Revision/Reinitialization.....TAS-20

    Data File Documentation.....TAS-20

SCREEN MASK UTILITIES.....TAS-23

APPLICATION INITIALIZATION (START) PROGRAM GENERATION.....TAS-25

APPLICATION MENU PROGRAM UTILITY.....TAS-26

DATA ENTRY/INQUIRY/UPDATE PROGRAM GENERATION.....TAS-28

REPORT/FORM PRINTNG UTILITIES.....TAS-31

SUPPLEMENTARY DATA FILE UTILITIES.....TAS-35

SYSTEM RESIDENT MACROS.....TAS-36

REVIEW QUESTIONS.....TAS-39

ANSWERS.....TAS-42

APPENDIX LIST OF SUBROUTINES AVAILABLE IN IDEAS SYSTEM

    SUBROUTINES MODULES.....TAS-43

**ABSTRACT**

Inquiry Data Entry Access System (IDEAS) is a relatively recent addition to Wang's software resources, especially in comparison with ISS and GBS. IDEAS was first released in 1980 as a powerful program development tool. Its primary purpose is to assist programmers by providing a skeleton program (or framework) around which sophisticated application programs can be built. Among the specific tasks that IDEAS greatly facilitates are:

- creation and maintenance of data files,
- generation of customized screen formats,
- creation of menus with or without password protection,
- generation of programs for data entry, inquiry, or update operations, and
- production of customized reports.

There are two IDEAS resources that should be frequently consulted by systems analysts. The first is the IDEAS data sheet (700-5747) which gives a quick overview of IDEAS and its capabilities. The data sheet is particularly useful when explaining the power of IDEAS to potential customers. The second document is the IDEAS User Manual which is the definitive technical resource on IDEAS. Specific questions on format, coding, and program conventions are answered in the User Manual.

This module on IDEAS provides you with a firm understanding of the purpose and features of the IDEAS package and its individual utilities. It will not, however, repeat the specific detail in the IDEAS User Manual. You can acquire detailed technical information from the User Manual and from your own experience with the package.

## INQUIRY DATA ENTRY ACCESS SYSTEM

---

### OBJECTIVES

At the conclusion of this module you will be able to:

- Define the purpose of IDEAS.
- Identify the intended audience for IDEAS.
- Define the hardware requirements for IDEAS.
- Describe the purpose, use, and examples of
  - data file utilities,
  - screen mask utilities,
  - start program generation,
  - application menu program utilities,
  - data entry/inquiry/update program generation,
  - report/form printing utilities.
- Describe the purpose and use of supplementary data file utilities and system-resident macros.

### MATERIALS REQUIRED

IDEAS User Manual - 700-5778  
IDEAS Data Sheet - 700-5747

### DIRECTIONS FOR COMPLETION

After completing this Module, contact the Course Administrator for the Module Test.

## INTRODUCTION

Inquiry Data Entry Access System (IDEAS) is a powerful development tool which can be used to create comprehensive application programs. GBS, as you remember, is a series of application programs; IDEAS is different in that it is a system to assist in the development of custom application programs. IDEAS generates a framework which can be adapted, expanded, and customized to meet each customer's data processing needs. This framework consists of such essential parts as establishing data files, regulating data entry, creating program menus, and determining the format of reports. Without IDEAS, these tasks can take a great deal of time and programming effort. With IDEAS, these jobs are greatly simplified. Thus, reduced time and programming effort are required to produce efficient and effective application programs.

The cornerstone of IDEAS is its six system utilities. The various utilities are listed and briefly defined below. They are treated in more depth later on in this module.

- Data Files Utilities - Data file definitions are created, revised, and documented through these utilities. Each data file must have one primary key associated with it.
  
- Screen Mask Utilities - These utilities provide a user with an easy to use tool for developing the screens necessary for interactive application systems. The user can create screens that simplify data entry and reduce the probability of incorrect entering of information.
  
- Application Initialization (START) Program Generation - A module, often called START, is generated from user input that initializes system addresses and operating parameters, and opens user-specified files.
  
- Application Menu Program Utility - Menu displays can be quickly created through this utility. Up to thirteen programs and/or menus can be called from this menu, and a password security option is available.

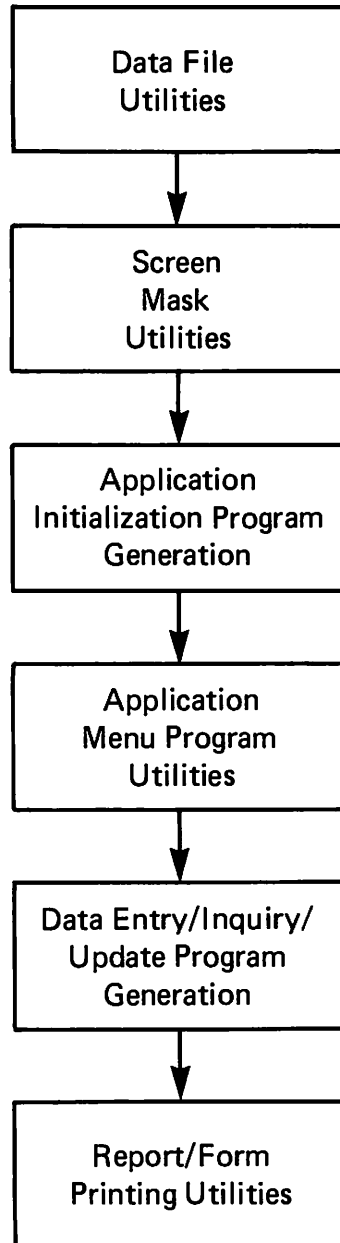
## INQUIRY DATA ENTRY ACCESS SYSTEM

- Data Entry/Inquiry/  
Update Program  
Generation - Eight different types of data entry programs can be generated by this utility. These data entry programs allow data manipulation on files created through the Data File Utilities.
- Report/Form Printing  
Utilities - The user is able to create report definition files which define the content and format of reports. Each report may access up to four data files.

The sequence of these six system utilities may vary to an extent. The only requirement is that the data file and screen mask utilities must come first. Because all other operations in IDEAS are based on data files being defined and the information being entered, these two utilities must precede all the others. After these two, the sequence may differ depending on the needs of the particular application. A typical order, however, is to develop the START module next. This module is part of the application program itself. In the START module, the user is asked to identify the program to load; this is usually a main menu which keys access to the rest of the application program. The main menu is created by the Application Menu Program Utility.

Following the development of the START module and the main menu, the data entry or access programs are generated; these programs include a choice of eight different inquiry, data entry, or update functions. If these data entry, inquiry, or update programs need to be expanded or altered, the modification can be accomplished at this time or after report generation. The final step is usually the creation of a customized report to output the results of the application program. This sequence of the six IDEAS System Utilities is presented in Figure 1 and it is the order in which the utilities are treated here. But, as previously mentioned, the last four steps in this sequence may vary depending on the nature of the job.

Figure 1. Typical Sequence of the Use of IDEAS Program Development Utilities





## FEATURES AND BENEFITS

As you will determine from reading about each of the six system utilities, IDEAS has a number of features which make it a very powerful and flexible system. Among the most important features are the following:

- Automatic file maintenance capability.
- Comprehensive data file utilities support the system.
- Password security provided.
- Modularized BASIC code generated for easy use and adaption.
- Menu-driven system with easy-to-use screen prompts.
- Series of system resident MACROS available to all IDEAS programs.
- Easily generated screens.
- Flexible report generator.

There are two additional features of IDEAS that are also very important. The first is that IDEAS is a self-documenting system. In many software packages, documentation is an afterthought or not provided at all. Providing documentation saves a great deal of time. In IDEAS, the important role of documentation is recognized and very easy to accomplish; it is a menu-selectable option for each IDEAS utility. The hard copy documentation provided by IDEAS is very useful in the input of information to other IDEAS utilities and in making program modifications. The second feature is that the IDEAS package is written with IDEAS. Programmers can see first hand how IDEAS works, for example, by examining the code that generates IDEAS screens, menus, and reports. Or, if a programmer wants to see how the IDEAS system resident MACROS are used, all he has to do is to examine an IDEAS listing because these DEFFN subroutines are used internally.

These IDEAS capabilities and features provide both the vendor and experienced applications programmer as well as the customer with several important benefits. Vendor or experienced programmers are usually the people who work with the IDEAS package to develop application software for the end user. IDEAS generates the skeleton programs around which they make appropriate

## INQUIRY DATA ENTRY ACCESS SYSTEM

modifications and enhancements. Programmers or vendors save considerable time because they can employ IDEAS file maintenance, MACROS, and other system support options, and because they do not have to generate all of the program code. The cost of software development is constantly rising, and thus IDEAS often can save the customer substantial sums of money by reducing program development time.

With this general understanding of the purpose, features, and benefits of IDEAS and its six system utilities in mind, we can now turn to a more in depth treatment of the IDEAS package. The first topic discussed is the hardware requirements for IDEAS. This discussion is followed by a closer examination of the six system utilities.

## HARDWARE REQUIREMENTS

IDEAS Release 1 can be run on 2200T, VP, SVP, MVP, and LVP systems. It requires a CRT with an 80 x 24 screen. IDEAS takes full advantage of the features on the 2236DE terminal (e.g., blinking, reverse video, and bright display) and consequently makes data entry easier for the operator. IDEAS is supplied either on two 2270 diskettes (one for development utilities and one for run time utilities\*) or on one 2270A diskette (both utilities combined). In addition to the diskettes for the utilities, a disk address must be specified for data file storage. It is recommended that IDEAS be moved to a hard disk system whenever possible so that access will be faster, storage space increased, and all parts of IDEAS can be located on one disk. A program called "ID MOVE" is provided on the utilities diskette to facilitate transfer of IDEAS to hard disk.

In terms of the amount of memory, single-user systems (2200T, VP, and SVP) require 32K to run IDEAS. In multi-user systems (2200 MVP, and LVP), a 30K partition is required for the utilities; in addition, a 17.5K global partition on each bank is needed for the run-time utilities.

One of the first actions taken in the startup of IDEAS is to select the disk address for the system utilities. The screen displays a number of allowable disk addresses and the user must select the disk address for the system utilities from this list. This screen is presented in Figure 2.

---

\* The run-time utilities are a group of programs that mainly perform file maintenance activities.

# INQUIRY DATA ENTRY ACCESS SYSTEM

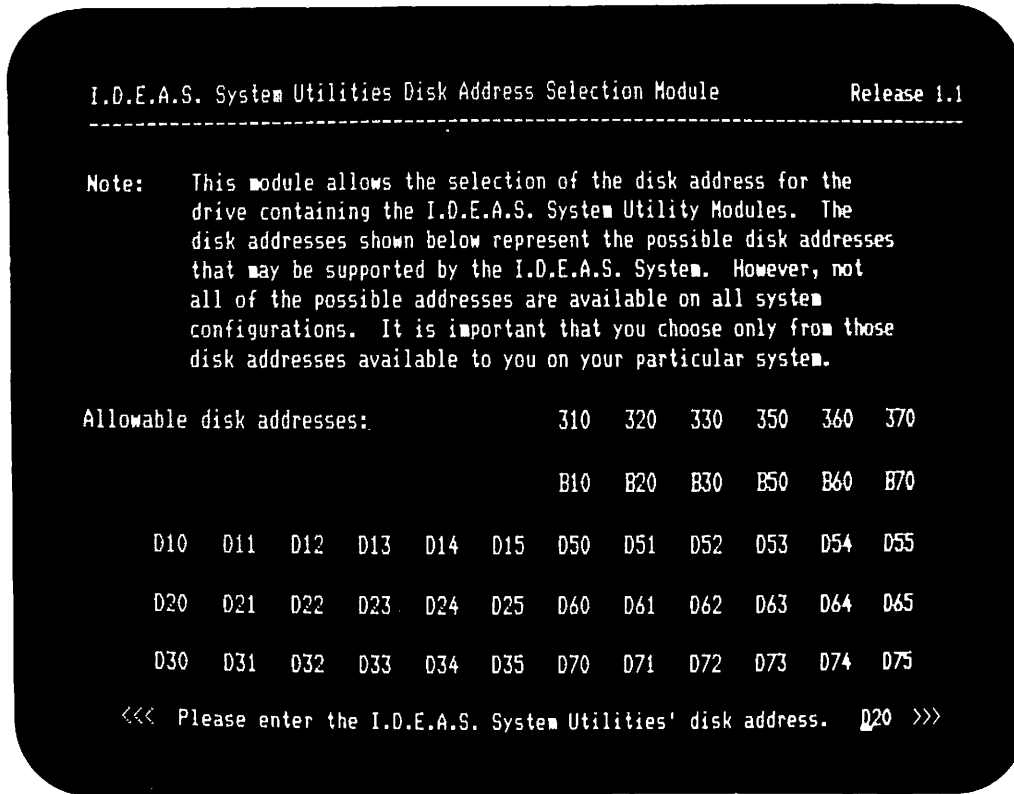


Figure 2. IDEAS System Utilities Disk Selection Screen

Notice that a default disk address is presented following the prompt "Please enter the IDEAS System Utilities disk address." If the default address is the proper choice, press RETURN.

The next screen that appears is also related to the hardware configuration for IDEAS. The Application Device Selection Module allows the user to set the disk addresses for the items listed on the screen. The addresses set in this module are those used in creating data files, screens, report masks, menus, START modules, and data entry programs. When configuring your own system, it may be different from the example cited.

```

I.D.E.A.S. System Utility - Application Device Selection Module      Release 1.1
=====
Device # 01 / 204 - Printer address ( 204 211 212 213 214 215 216 )
Device # 02 / D20 - Disk address for I.D.E.A.S. System Utilities
Device # 03 / D21 - Disk address for application screen or report mask files
Device # 04 / D21 - Disk address for application program files
Device # 05 / D21 - Disk address for data record definition files
Device # 06 / D21 - Disk address for application data files
=====

Note: Application data file disk addresses below are for 2200 VP & MVP only.
=====
Device # 07 / D20 - Disk      Device # 10 / D20 - Disk      Device # 13 / D20 - Disk
Device # 08 / D20 - Disk      Device # 11 / D20 - Disk     Device # 14 / D20 - Disk
Device # 09 / D20 - Disk      Device # 12 / D20 - Disk     Device # 15 / D20 - Disk
=====

Allowable disk addresses: ( Not all addresses may be available on all systems. )
(2200 T, VP, & MVP ) 310 320 330 350 360 370
                    B10 B20 B30 B50 B60 B70
(2200 VP & MVP only) 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
=====
Touch EXEC to accept as is, or SF Key corresponding to device # to be changed.

```

Figure 3. Application Device Selection Module

With configuration complete, IDEAS can be run. The IDEAS System Utilities Menu is presented in Figure 3. Notice that FN 05 to 11 are the spaces reserved for the six utilities. Additional menu choices can be entered in the other available positions, but FN 05 to 11 are always reserved for the system utilities.

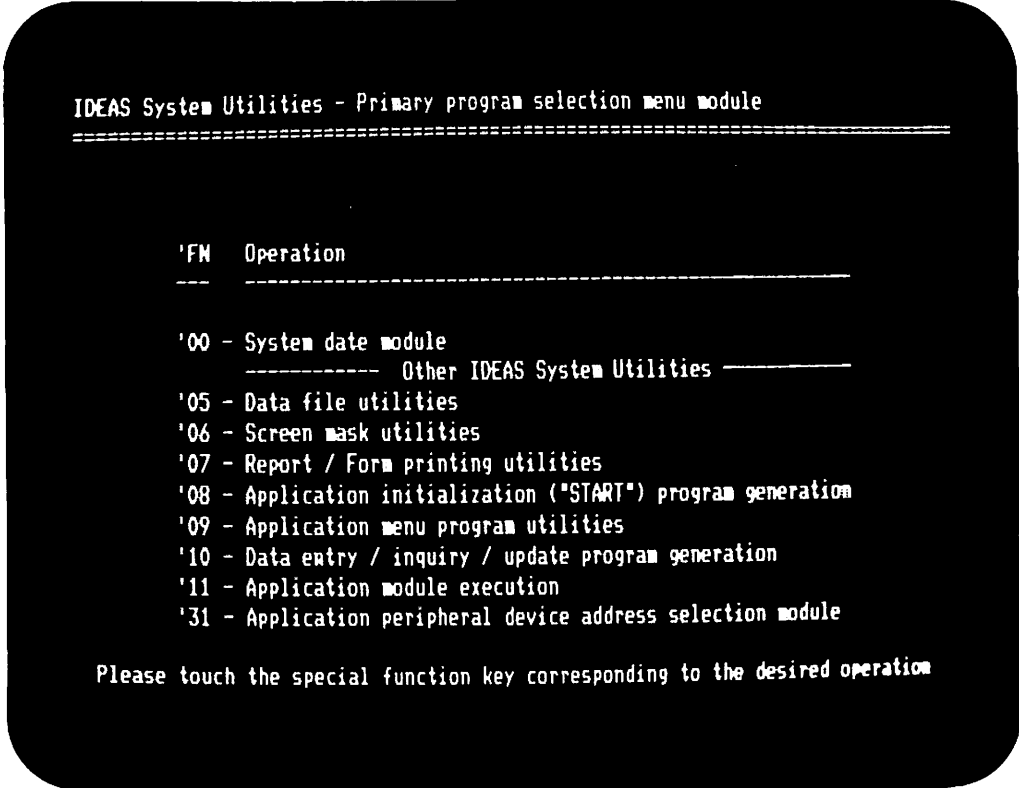


Figure 4. Ideas System Utilities Menu

## DATA FILE UTILITIES

The IDEAS Data File Utilities are used to define, initialize, and document all primary data files and alternate key files used by the system. The utilities may also be used to alter parameters in these files and re-initialize them after changes have been made.

Each data file must have one primary key associated with it. Up to sixteen alternate key files may be associated with each primary data file. Primary keys are stored in the data file itself, while alternate keys are placed in separate files. These alternate key files contain the keys and pointers necessary to manipulate the data in the primary data file for reporting, data entry, inquiry, or modification. Two types of duplicate alternate keys are available in Release 1--one which optimizes sequential processing, and another which increases efficiency in a random access environment. All key file maintenance, regardless of file type, is automatically performed by the system.

The IDEAS file management system is HIKAM: Hashed Index Keyed Access Method. HIKAM combines hashing and indexing techniques to provide excellent performance in both random and sequential access environments. HIKAM's unique blend of is significantly faster than other access methods. HIKAM files are formatted under software control, to optimize search and access time and to provide efficient disk management.

There are three separate menu selections in the Data File Utilities. Each has its own purpose and use. The three menu choices are:

- New Data File Creation
- Existing Data File Revision/Reinitialization
- Data File Documentation

### New Data File Creation

**Purpose:** The file creation module allows the user to specify the name and define the parameters of a new primary data or alternate key file. The user is taken through the module on a screen-by-screen basis. When the record has been described, the keys selected, and the file's construction and location defined, the file may be initialized. After initialization, the file can be used by other IDEAS modules. This is the first utility run when a development of a new application program is begun.

## INQUIRY DATA ENTRY ACCESS SYSTEM

**Use:** This module must be entered to create any IDEAS file. Therefore, it is used frequently in program development operations.

**Examples:** There are a number of different screen-based operations in the New Data File Creation Module. These include:

- Name specification screen.
- Primary address/file type selection screen.
- Data record field definition screen.
- Key field selection screen.
- Data file performance option selection screen.
- Data file initialization, and
- Similar procedures for alternate Key File creation.

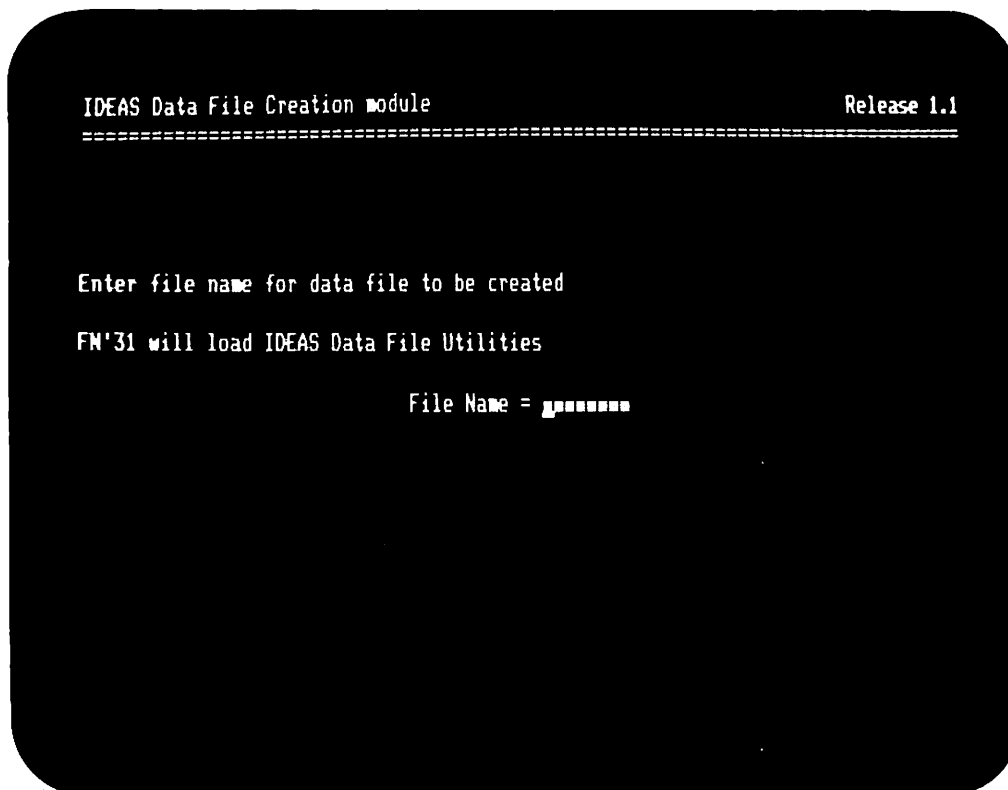


Figure 5. Name Specification Screen



Name Specification Screen - The user enters the file name, which may be from 1 to 8 characters.

Primary Address/File Type Selection Screen - After a primary file name has been specified, this screen requests two additional pieces of information. The first is the disk address for the file, and the second is the file type. The file type is a single digit keyed to the table displayed on the screen. The key variables in file type are whether the file is a primary or alternate, and whether alternate or duplicate keys are allowed.

```

IDEAS Data File Creation/Re-initialization module
                                                    Release 1.0
-----
File name = "BCTEST "   Disk address for this file = E20   File type (1-6) = 1
Associated primary file name (alternate key or key/data files only) = "BCTEST "
Available file types:

```

Type Description	Data record location	Allow Duplicate Keys ?	Allow Alternate Keys ?
1. Primary key/data file	Data segment in file	No	Yes
2. Primary key/data file	Data segment in file	Yes (adjacent)	Yes
4. Alternate key file	Type 1 or 2 data segment	No	No
5. Alternate key file	Type 1 or 2 data segment	Yes (adjacent)	No
6. Alternate key file	Type 1 or 2 data segment	Yes (scattered)	No

```

-----
Attention      Touch EXEC to accept as is, EDIT to modify      □

```

Figure 6. Primary Address/File Type Selection Screen

INQUIRY DATA ENTRY ACCESS SYSTEM

Data Record Field  
Definition Screen -

The user must enter the names, lengths and types of all fields in the data record. This screen appears after the file type and location have been specified. This screen displays the fields in alphabetical order and keeps a cumulative total of the number of fields, record length, and the number of bytes packed for both numeric and uppercase alphabetic types.

```
IDEAS Data File Record Field Definition Module          Release 1.1
=====
File name "VENDFILE"  Disk address = D21
No. of fields = 7 Length = 86 # @ 2:1 = 10 # @ 4:3 = 16 Packed = 78
-----Field Names-----
ADDRESS
CITY
NAME
PHONE
STATE
VENDNUMB
ZIP CODE

-----
Field name = "      " Length (1-64) =      Type (1=num 2=UC a/n 3=any a/n) =
```

← Cumulative Totals

↑  
Enter Field  
Name Here

Figure 7. Data Record Field Definition Screen

Key Field Selection  
Screen -

Once the user has defined all fields, the key must be specified. The key is composed of one, two, or three fields within the data record. The key cannot total more than 58 bytes. The user enters the name of the key field and then either a + or - to indicate whether the key is to be sequenced in ascending or descending order.

Key Field is  
Entered Here

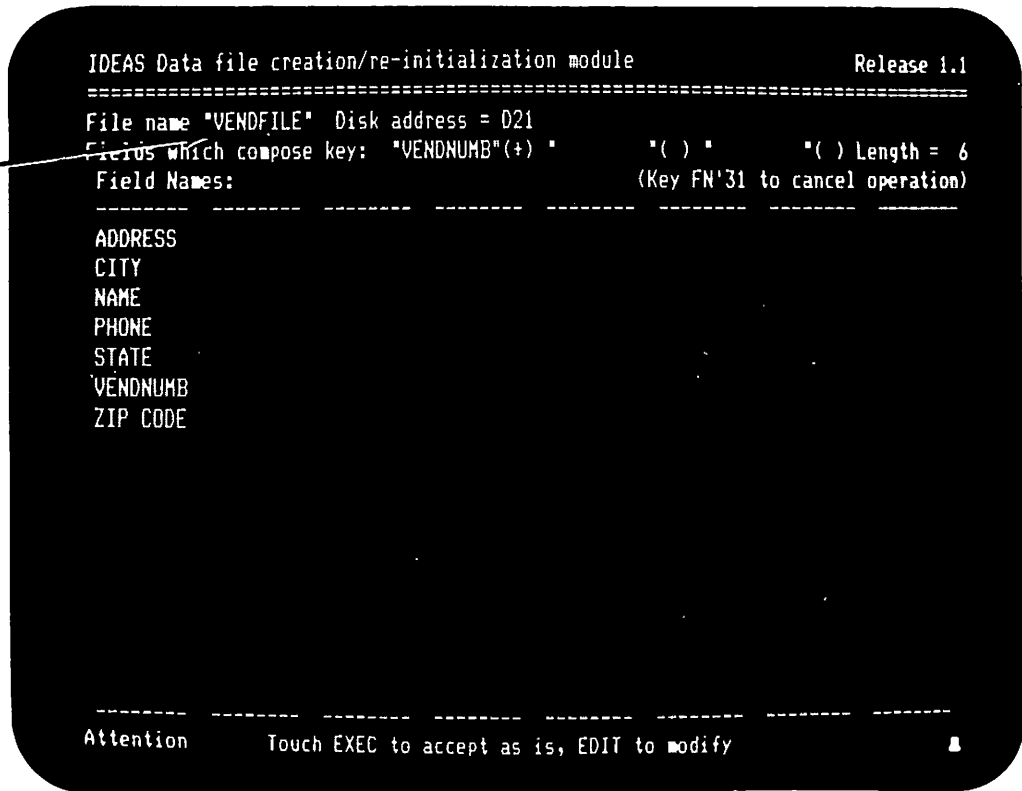


Figure 8. Key Field Selection Screen

**INQUIRY DATA ENTRY ACCESS SYSTEM**

Data File Performance  
Option Selection Screen -

This screen allows the user to customize the file by specifying the number of records, the number of disk volumes the file is to span, and the type of record blocking. There are three options in record blocking: one that provides the fastest access time, one that provides the most efficient use of disk space, and one that is a mixture of access and efficiency. After a choice of record blocking is made, seven file structures are displayed. The user can select the desired file structure; this decision should depend primarily on whether the file access normally will be random or sequential.

IDEAS Data File Performance Option Selection Module										Release 1.0
File Name =	"BCTEST"	Type 1 (Primary)	No. of records (specified) =							100
Record Length =	49	# of vols. = 1	Opt#1 = 2	Opt#2 = 3	(actual) =				105	
Packed numeric =	20	Vol/Adrs 1/	2/	3/	4/	5/	6/	7/	8/	
Packed alpha =	24	Available								
Packed length =	34	Required								
Key length =	2	Desired								
-----										
Option # 1 ( Record blocking )	Rec / Sec	XtraPad	XtdLen	%Waste	Time					
1. Fastest access time	7 / 1	2	36	7.031	150					
2. Best time/disk compromise	7 / 1	2	36	7.031	150					
3. Most efficient disk use	15 / 2	0	34	0.391	152					
Option # 2	---RANDOM---		-----SEQUENTIAL-----				---DISK---			
File structure	ins/del	rtv	1st >=K	First	Next	Memory	Sectors			
adjustment option	(ms)	(ms)	-(sec)-	-(sec)-	(ms)	(bytes)	Needed			
1. Fastest random access	150	150	0.250	0.175	100	32	24			
2. -----	183	150	0.150	0.125	100	16	22			
3. -----	183	150	0.150	0.125	100	16	22			
4. -----	183	150	0.150	0.125	100	16	22			
5. -----	183	150	0.150	0.125	100	16	22			
6. -----	183	150	0.150	0.125	100	16	22			
7. Optimum sequential	183	150	0.150	0.125	100	16	22			

Figure 9. Data File Performance Option Selection Screen

Data File Initialization - The final step in the creation of a data file is the initialization process, where disk space is allocated for a file. When the screen appears, the default file name is the one defined in the Name Specification Screen. Any associated files (alternate key files for the primary file) are also displayed and may be initialized along with the default file. You do not have to initialize the file at this time, but instead can initialize primary and all alternate files after the final file has been created.

```

IDEAS Data File Utilities - Data File Initialization Module          Release 1.1
=====
Note: Initializing file will destroy all data and/or keys currently in the file.
Touch EXEC to initialize file "VENDFILE", EDIT for another file, FN'31 to cancel
EXEC: Primary file, FN'0 - FN'15: Alternate key file, FN'16: All files shown.<>
=====

Now Initializing file "      "          Initialization is      % complete

Number of buckets in the file =          Now processing bucket number
Number of index sectors/bucket =         Now processing index sector #

=====
EXEC = Primary data file "VENDFILE"      FN'08 = Alternate key file "
FN'00 = Alternate key file "             FN'09 = Alternate key file "
FN'01 = Alternate key file "             FN'10 = Alternate key file "
FN'02 = Alternate key file "             FN'11 = Alternate key file "
FN'03 = Alternate key file "             FN'12 = Alternate key file "
FN'04 = Alternate key file "             FN'13 = Alternate key file "
FN'05 = Alternate key file "             FN'14 = Alternate key file "
FN'06 = Alternate key file "             FN'15 = Alternate key file "
FN'07 = Alternate key file "             FN'16 = Primary & all alternate files
=====
    
```

→ If more than one file is being initialized, this status display is helpful

Figure 10. Data File Initialization Screen

## INQUIRY DATA ENTRY ACCESS SYSTEM

### Existing Data File Revision/Reinitialization

**Purpose:** After a data file has been defined, it may be revised using this program. This program proceeds from screen to screen exactly as if the file were being created, except that each screen is displayed with information already filled in. The user then can accept or edit each screen and each entry in that screen as it is displayed. If a file has been initialized and changes need to be made, the file must be re-initialized after the alterations have been made. In the case of a primary data file, all associated alternate key files must also be re-initialized.

**Use:** This module is important because revision of data files is a continuing process. Without provisions for updating and revising files, the IDEAS Utilities would be more cumbersome for the programmer.

**Examples:** The screens that appear when this Revision/Reinitialization option is chosen are the same as those in the Data File Creation module. In most screens there is a prompt which states "Touch RETURN to accept as is, EDIT to modify." Changes can be made by pressing EDIT and entering the changes. On some screens, special function keys are also part of the revision process. For example, the following appears in the Data Record File Screen: RETURN=New field, EDIT=Revise, FN'09=Delete, FN'20=Exit field edit mode.

### Data File Documentation

**Purpose:** Documentation can be produced on any initialized data file at any time. Three separate screens are provided; in addition, a single page can be printed which highlights the information on all three screens. Documentation of each file is important because it is needed in construction of related files, screen masks, and application programs.

**Use:** This program is valuable because it can be difficult to keep an accounting of all the various data files, their locations and structures. The documentation screen provides this information in a readable format.

Examples: After the name of the file is entered, a screen is presented which lists file name and type, number and address of volumes used, number and packed length of fields, key fields and length, position and type, performance and blocking options selected, all related alternate key files, and the date of last revision. As stated previously, this type of documentation is very valuable in further applications development (e.g., creating screen masks). This screen is displayed in Figure 11.

```

IDEAS Data File Specification Record Layout Documentation Review      Release 1.0
-----
Logical File Name = "BCTEST "                                     Number of fields          7
Physical File Name = "BCTEST "                                    Record length             49
Primary File Name = "BCTEST "                                     # packed numeric          20
File Type = 1                                                    # packed alpha            24
                                                                    Packed length             34
                                                                    Last revision date        061000

Number of volumes = 1
Volume # 1 Disk Address = E20
Volume # 2 Disk Address =
Volume # 3 Disk Address =
Volume # 4 Disk Address =
Volume # 5 Disk Address =
Volume # 6 Disk Address =
Volume # 7 Disk Address =
Volume # 8 Disk Address =

Option # 1 (record blocking)= 2
Option # 2 (performance) = 3

Associated Alternate Key Files:
1. "BCTEST2 " 9. "
2. " " 10. "
3. " " 11. "
4. " " 12. "
5. " " 13. "
6. " " 14. "
7. " " 15. "
8. " " 16. "

Total key length = 2 Key 1 Key 2 Key 3
Field name ID-MUM
Order +
Position 001
Length 02
Type 1
Attention Touch EXEC to continue to next screen or FN'31 to cancel
    
```

Figure 11. IDEAS Data File Specification Record Layout Documentation Review

**SCREEN MASK UTILITIES**

**Purpose:** The Screen Mask Utilities provide the user with an easy-to-use tool for developing the screens necessary for interactive application systems. These screens define a formatted display that facilitates operator entry of information for the purpose of creating or updating a data file. A screen mask may contain operator instructions, headings, data field descriptions, or any special display that will assist the operator in entering the correct data. The screen mask created by these utilities is displayed by the Data Entry/Inquiry/Update IDEAS Utility.

There are three Screen Mask Utilities. The first is a Screen Mask Definition Module in which the operator enters the information needed and its format on the screen. The operator has complete freedom to arrange the screen in the manner that is most compatible with his data processing needs. Each field on the screen is then defined in terms of such factors as length, type, optional or required status, and acceptable values. This screen mask information is saved in a definition file that is used by the IDEAS Data Entry Utility. The second Screen Mask Utility is used to revise existing screen masks. This utility goes through the same steps as the Definition Module except that instead of the blank screen that appears in the Definition Utility, the existing screen mask is displayed in the Revision Utility. The third Screen Mask Utility is a documentation module which can print a hard copy of the screen mask and the list of attributes for each data field.

**Use:** Screen masks make data entry more efficient and reduce the chance of error. Screen masks are especially important for computer operators who are not very familiar with data processing and file maintenance procedures. Screen masks make data entry much easier and more logical for many computer operations. This IDEAS Utility allows the user to create customized screen masks very quickly and with very little operator effort.

**Example:** The first display in the Screen Mask Definition Utility asks for the name of the data file to contain the screen mask definition. Then, the name of the file that contains the fields that are masked on the screen is entered. Next, the Utility displays the Mask Editor Screen; in the case of a new screen mask, the screen is blank. The operator then enters text, comments, heading, or other information in any





```

I.D.E.A.S. System Utility - Create / Revise Screen "PSCRN"      Release 1.1
=====
'FN Field parameter -----
'00 Field name ? ..... "UNITCOST" Valid Character List
'01 Row on screen ? ..... 15      1 Digits only
'02 Column on screen ? ..... 14    2 Digits & decimal pt
'03 Position in Record ? ..... 10  3 Digits & signs
'04 Default field ? ..... No default field 4 Digits, signs, & dec
'05 Field length ? ..... 8        5 Upper case letters
'06 Valid characters ? ..... Digits, signs, & dec 6 UC alpha & digits
'07 Allow keyboard entry ? ..... Yes 7 UC, digits, & punct
'08 Allow display ? ..... Yes     8 Any character
'09 Required or optional ? ..... Required field 9 FN Keys, EDIT & EXEC
'10 Full if present ? ..... Need not be filled
'11 Left or right justified ? ..... Right justified
'12 Zero or space fill ? ..... Space filled
'13 Number of decimal places ? ..... 2
'14 Termination Full/EXEC ? ... Terminate when full
'15 Save field parameters
-----
'25 Delete current field '31 Cancel to mask editor
'26 Insert current field  EDIT Display screen mask

Touch the FUNCTION KEY corresponding to the desired operation.
    
```

Figure 13. Field Parameter Selection Screen

The last step after defining all fields on the screen is to save the screen mask. This action is taken by pressing FN'20 from the Mask Editor. The IDEAS User Manual deals with two special considerations on the Screen Mask Utilities. The first consideration is when fields are inserted or deleted when default field values are used; and the second is when fields from several files are combined in the screen display. Refer to Chapter 6 in the User Manual when these considerations arise.

**APPLICATION INITIALIZATION (START) PROGRAM GENERATION**

**Purpose:** This utility creates and revises modules that are used to begin program operations; these modules are termed START modules because their function is literally to start the application program. The START module initializes system addresses and operating parameters and opens user-specified files. It then loads a user specified program, which is usually a main menu to govern the flow of the program.

**Use:** After data files have been defined and initialized, it is necessary to create a module which begins program operation. This utility performs this common function quickly and efficiently.

**Example:** The START module screen asks the user to enter the name for the load program, the load message, the system device address, and several available options. It is presented in Figure 14. The next screen in this module requests a list of data files that will be used in the application program.

Main menu.  
that is called  
by START 01  
(see next utility)

```

IDEAS System Utilities - "START" Program Creation/Revision Module   Release 1.1
=====
Program Name: "START01 "                Last revision date: 072980 ( JUL 29 80 )
Load Program: "MMENU "
Load Message: "NOW LOADING VENDOR PROGRAM MAIN MENU"
=====
Peripheral Device Addresses:
Printer ----- # 01 / 204
IDEAS Utility Disk ----- # 02 / D20
Screen/Report Mask Disk ---- # 03 / D21
Application Program Disk ---- # 04 / D21
Data File Description Disk -- # 05 / D21
Application Data File Disk -- # 06 / D21
* Application Data File Disk -- # 07 / D20
* Application Data File Disk -- # 08 / D20
* Application Data File Disk -- # 09 / D20
* Application Data File Disk -- # 10 / D20
* Application Data File Disk -- # 11 / D20
* Application Data File Disk -- # 12 / D20
* Application Data File Disk -- # 13 / D20
* Application Data File Disk -- # 14 / D20
* Application Data File Disk -- # 15 / D20
* Devices #07-#15 not for use on 2200T CPU

Available Options:
1. "CANCEL" FN Key number      31
2. Skip ahead keys on (Y/N) N
   ( FN'4, FN'11, FN'12 )
3. Skip back keys on (Y/N) Y
   ( FN'7, FN'13, FN'14 )
4. System error messages (Y/N) Y
5. System error msg trap (Y/N) N
   ( DEFFN '99 )
6. Function key trap (Y/N) N
   ( DEFFN '98 )
    
```

Figure 14. START Module Screen

# INQUIRY DATA ENTRY ACCESS SYSTEM

## APPLICATION MENU PROGRAM UTILITY

**Purpose:** This IDEAS Utility creates, revises, and documents menus. Up to 13 submenus (programs) can be called per menu. The Menu Program Utility is very easy to use. The operator enters the name of the menu to be created, the load message to be displayed as the menu is loaded, and the menu program file names. A password security option is available for menus which should have limited access.

**Use:** Menus are, of course, very important to the operation of an application program. This Utility allows menus to be created quickly and effectively.

**Example:** Two examples of this utility are included. The first is a screen in which the operator creates or revises the menu. The program name, loading message, file names called from the menu, and their load message are entered on this screen. The second is the screen that provides printed documentation for the created menu. Each is shown:

```
I.D.E.A.S. System Utility - Menu Generation/Revision Module      Release 1.1
-----
Program Name: MMENU
Load message: Loading the MAIN MENU

Key  Program  Description on menu / load message
-----
'X1  STOP      Return to 'IDEAS' Application Peripheral Selection Module
'00  VMPROG     VENDOR MAINTENANCE PROGRAM
'01  PMPROG     PRODUCT MAINTENANCE PROGRAM
'02  PRODREPT  PRODUCT LISTING BY VENDOR
'03  IDFU-340  TC MENU
'04
'05
'06
'07
'08
'09
'10
'11
-----
Password ( if any ) to be required at run time for use of this menu: "
Attention      Touch EXEC to accept as is, EDIT to modify
```

↑  
Password security option  
is entered here

Figure 15. Menu Program Utility Screen

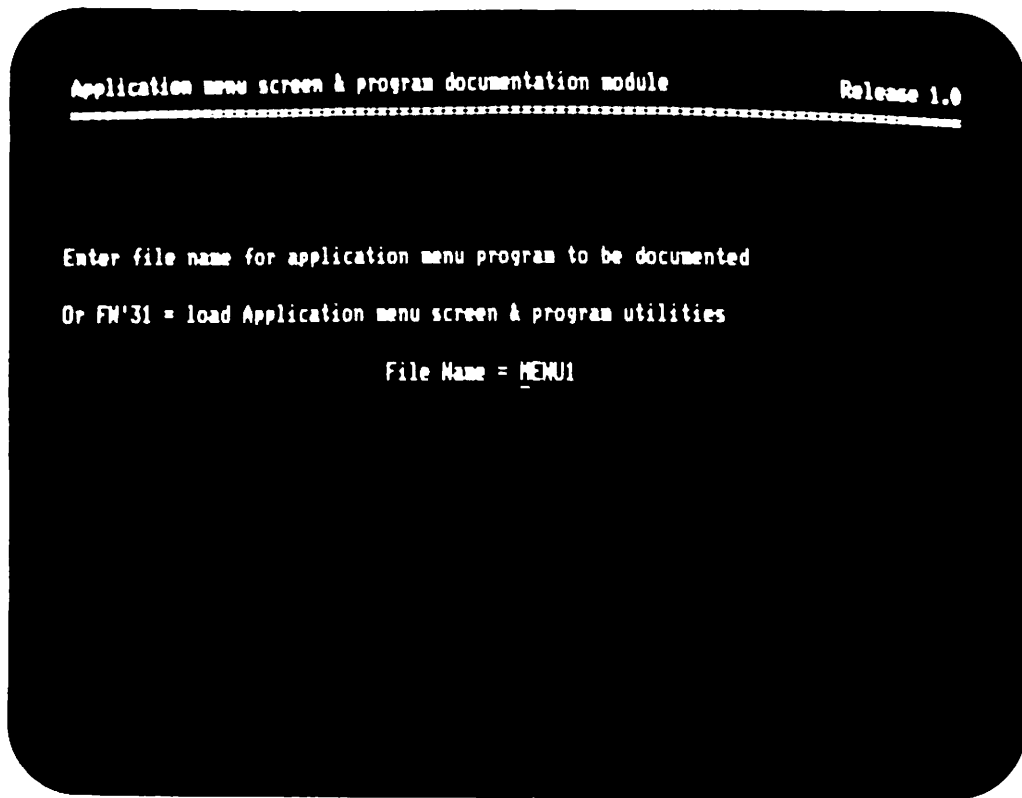


Figure 16. Menu Program Documentation

## DATA ENTRY/INQUIRY/UPDATE PROGRAM GENERATION

Purpose: Data entry, inquiry, and update programs allow data manipulation on files created through the IDEAS Data File Utilities. To create a data entry program, the user need only enter the name of the program to be created, the screen format, and data file to be used, and specify the type of data entry program desired. All of this information is entered on one screen; in effect, this powerful utility and the code that it generates all stem from just one screen. There are eight types of data entry programs that are supported by this utility:

1. Inquiry only, i.e., information is only displayed; it cannot be modified.
2. Add a new record to the file.
3. Add a new record or modify an existing one.
4. Add a new record or delete an existing one.
5. Add a new record or modify or delete an existing one.
6. Modify an existing record.
7. Delete an existing record.
8. Modify or delete an existing record.

Each of these data entry programs has a different set of available data entry operations. A password protection menu option is available to help guarantee the security of sensitive information that should not be widely disseminated. The screen mask generated by another IDEAS utility is used in the Data Entry/Inquiry/Update Program Generation Utility. The Data Entry Programs generated by this utility are designed to access one file. The programs will need to be modified when access to more than one file, or more than one screen per program, is required. These are the same special conditions that exist for the Screen Mask Utilities. But these changes are fairly easy to accomplish because the BASIC code generated by the Utility is modularized and designed for potential user adaptation. The IDEAS User Manual presents three examples of user modification to data entry programs.

**Use:** A data entry program must be created for each data entry function desired. This utility is frequently used, especially in situations where a company constantly needs to update its business records, product listings, or payroll, or to perform other similar jobs.

**Example:** As previously mentioned, this utility is based on one screen. And as can be seen in Figure 16, the screen is simple and straightforward.

```

IDEAS System - Data Entry/Inquiry/Update Program Generation Utility Release 1.1
=====
Create/revise date: JUL 29 80
File name for program to be generated: *UPDATE *
File name for screen mask to be used: *PSCRM *
File name for data file to be accessed: *PFILE *

Functions to be provided in this program ( choose one from the list below ): 5

1. Inquiry only
2. Add a new record to the file
3. Add a new record or modify an existing one
4. Add a new record or delete an existing one
5. ==> Add a new record or modify or delete an existing one
6. Modify an existing record
7. Delete an existing record
8. Modify or delete an existing record

Number of fields required on screen to establish the key to the data file: 1

Number of extra disk sectors to be provided for additional application code:
    
```

**Figure 17. Screen for Data Entry/Inquiry/Update Program Generation Utility**

Figure 18 presents an example of a program (VENDPROG) that is generated by this utility. This program is designed to add a new record, or modify or delete an existing one (choice 5 on the screen shown in Figure 17). The field names and other information are taken from values entered in Data File Creation and other utilities. The DEFFN statements in lines 1600 and 1800 are examples of the use of IDEAS resident MACROS that are available to any application program. In many cases, programs generated by this utility--such as VENDPROG--will be further expanded or modified depending on the data processing needs of the customer.

VENDPROG As Generated By IDEAS

1000	REM "VENDPROG" FEB 28 80 TYPES
1010	IF 00\$(9) ["M" THEN 1020: LOAD DC T#2, "IDGLBSSEL" 1010, 1010
1020	COM VO, F0\$8: F0\$="VENDFILE"
1030	GOSUB '32("VENDSCRN")
1040	REM Defaults
1400	GOTO 2000
1600	DEFEN'98: RETURN: REM FNKEY Trap
1800	DEFEN'99: RETURN: REM Error Trap
2000	F=0
2010	F=F+1
2020	IF F]FO THEN 4000
2030	GOSUB '34(F)
2050	ON FGOTO 2110, 2120, 2130, 2140, 2150, 2160
2110	REM "VEND#" " - 001: GOTO 3900
2120	REM "NAME" " - 002: GOTO 2010
2130	REM "ADDRESS" " - 003: GOTO 2010
2140	REM "CITY" " - 004: GOTO 2010
2150	REM "STATE" " - 005: GOTO 2010
2160	REM "ZIP CODE" " - 006: GOTO 2010
3900	GOSUB '82(F0\$)
3910	VO=V
3920	IF F\$(V) [" " THEN 2010
3930	GOSUB '86(-V)
3940	GOSUB '58(ABS(V), 0, K\$(1), 0)
3950	IF Q=O THEN 2010
3960	GOSUB '141(F0\$, K\$(1), 1)
3970	IF Q]L THEN 2030
3990	GOSUB '36
4000	E\$="Touch EXEC to accept, EDIT to modify"
4010	IF F\$(V0)=" THEN 4030
4020	STR(E\$, 37)=", FN'9 to delete"
4030	GOSUB '53(E\$)
4040	GOSUB '34(129)
4050	IF Q=24O THEN 2000
4060	IF Q=9 THEN 4140
4070	IF Q]32 THEN 4000
4080	GOSUB '142(F0\$, 1)
4090	IF Q]1O THEN 1030
4100	INIT(09)E\$
4100	STR(E\$, 35)="EDIT to modify, or CANCEL"
4120	GOSUB '35(E\$)
4130	GOTO 4040
4140	IF F\$(V0)=" THEN 4000
4150	INIT(20)Z\$()
4160	GOSUB '142(F0\$, 0)
4170	GOTO 1030

Figure 18. A Program Generated by the Data Entry/Inquiry/Update Program Generation Utility



## REPORT/FORM PRINTING UTILITIES

**Purpose:** This IDEAS Utility allows the user to create the content and format of reports. Complex reports can be designed quickly, with a minimum of operator effort. Each report may access up to four data files. The format specifications that can be designed in the Report/Form Printing Utilities include: report title, page headers, page footers, field sequence, and field spacing. Among the possible content specifications are the designation of new fields for reporting purposes, a sequence file for sorting, flags on which fields to print, and 32 different arithmetic operations. As with many IDEAS Utilities, there are actually three different Report/Form Printing Utilities: creation, revision and documentation. The difference between the first two is that in the revision program the previously entered values are displayed. The documentation module supplies the user with a printed, hard copy of the report and its specifications.

**Use:** This utility gives the user the opportunity to design customized reports without performing complicated programming operations. IDEAS does all the programming work and consequently a great deal of time, money, and effort are saved.

**Example:** As with several IDEAS utilities, a few screens are the key to an entire set of operations. In the Report/Form Printing Utilities there are two primary screens: Report Definition and Report Mask Editor. There is also the Report Documentation Module. An example of a Report Definition Screen is displayed in Figure 19. Table 1 defines many of the entries on the screen.



Table 1. Explanation of Entries for Report Definition Screen

Page header:	<p>The report title specified by the user. The page header appears at the top of each page of the report.</p> <p>If line one of the report mask contains any data or text, the system assumes this to be the page header; and, if a non-zero page length is specified, this line will be printed at the top of each new page.</p> <p>If the page header line contains the text MMDDYY or MMM DD YY, the date will be printed in that format in its place.</p>
Constants:	<p>Up to 10 constants may be specified for math functions – up to 8 numeric characters (0 to 9 and +, =, and decimal point).</p>
Default key range:	<p>A default low and high key may be specified as limits on the sequence file. These may be optionally modifiable by the user at run time.</p>
Logical record masks:	<p>Up to 4 fields may be tested for each record read. Only those records which meet all specified conditions will be included in the report.</p>
Group header: (optional)	<p>That set of line(s) which comprise a heading for a particular group (can consist of text and/or fields). The group header is printed once, and is printed again <u>only after a group field break or the beginning of a new page is encountered.</u> Group field breaks are inserted during report mask definition.</p>
Record header: (optional)	<p>A range of lines on the report mask consisting of text and/or fields that will be printed after the group header, then only after a record break field has been encountered. Record break fields are inserted during report mask definition.</p>
Record item:	<p>A range of lines on the report mask consisting of text and/or fields that will be printed for each valid record found in processing the sequence key file and passing the logical record mask tests.</p>
Record footer: (optional)	<p>A range of lines on the report mask consisting of text and/or fields that will be printed when a record break field is encountered. After printing, any math fields will be set to zero (record level sub-totals). The record footer follows the record header in report mask definition.</p>
Group footer: (optional)	<p>A range of lines on the report mask consisting of text and/or fields which will be printed only when a group break field is encountered. After printing, any math fields in this range will be set to zero (group level sub-totals).</p>
Report footer: (optional)	<p>A range of lines on the report mask consisting of text and/or fields which will be printed at the end of the report (report totals). The report footer is specified during report mask definition.</p>
Page footer:	<p>If a non-zero page length is specified, and if any text and/or fields are specified on the last two lines of the report mask, these will be printed on the last two specified lines of the page. For example, assume a 132 column report with a report mask length of 40 lines and a specified page length of 60 lines. Lines 39 and 40 of the report mask will be printed at lines 59 and 60 of each page. Note: Math fields appearing on these lines will be set to zero after printing each page (page totals).</p>
Math operations:	<p>Up to 32 math operations may be specified to be performed for each record processed.</p>

INQUIRY DATA ENTRY ACCESS SYSTEM

The Report Mask Editor (just as the Editor in the Screen Mask Utilities) allows the user to design a customized report. In the following example, a product listing by vendor is created. On the top part of the report the vendors number, name, and address appear. This is followed by a listing of all products sold by that vendor. This report uses the math functions option in the places where "totals" are requested.

LIST OF PARTS ON-HAND (BY VENDOR)		AUG 18 80 Page 001	
VENDOR NUMBER	00001	AMERICAN AUTO ACCESSORIES 35 PARK LANE NASHUA NH 07865	
PARTS SUPPLIED BY THIS VENDOR			
1	00100	13" WIPER BLADE REFILLS	QUANTITY = 25 @ 1.89 = 47.25
2	00101	FM RADIO ANTENNA	QUANTITY = 20 @ 9.00 = 180.00
3	00102	CHAMPION SPARK PLUGS (#7)	QUANTITY = 85 @ 1.20 = 102.00
4	00103	MOBIL ONE ENGINE OIL	QUANTITY = 48 @ 0.97 = 46.56
TOTAL VALUE OF ALL PRODUCTS SUPPLIED BY THIS VENDOR.....			<u>375.81</u>
VENDOR NUMBER	00002	MASON AUTOMOTIVE 8907 MIDDLESEX TPKE BURLINGTON MA 01885	
PARTS SUPPLIED BY THIS VENDOR			
5	00200	TRAILER HITCH (FORDS)	QUANTITY = 24 @ 48.25 = 1158.00
6	00201	TURTLEWAX LIGHT COMPOUND	QUANTITY = 18 @ 2.50 = 45.00
7	00202	ADJUSTABLE ROOF RACK	QUANTITY = 12 @ 26.80 = 321.60
8	00203	SUN TACHOMETER	QUANTITY = 12 @ 31.99 = 383.88
TOTAL VALUE OF ALL PRODUCTS SUPPLIED BY THIS VENDOR.....			<u>1908.48</u>
VENDOR NUMBER	00003	J & L DISTRIBUTORS, INC. 16 WEST MAIN STREET WESTFORD MA 01886	
PARTS SUPPLIED BY THIS VENDOR			
9	00300	RADIATOR CAP	QUANTITY = 18 @ 3.69 = 66.42
10	00301	TORQUE WRENCH	QUANTITY = 6 @ 22.50 = 135.00
11	00302	PUROLATOR AIR FILTERS	QUANTITY = 50 @ 4.85 = 242.50

Figure 20. Example of a Screen Created In Report Mask Definition Module

The IDEAS System Utilities are extremely powerful development tools that primarily assist the programmer by creating a framework for development of comprehensive application programs. The IDEAS System Utilities that already have been described are the heart of the IDEAS software package; however, there are two additional parts to IDEAS that should also be discussed. The first is the Supplementary Data File Utilities and the second is the System Resident MACROS.

#### SUPPLEMENTARY DATA FILE UTILITIES

These utilities, also known as Run-Time Utilities, are contained on the IDEAS application utilities diskette. (The previous six utilities are on the system utilities diskette.) The Supplementary Data File Utilities consist of a menu module and utility programs which provide specific user function. This menu can be accessed through any IDEAS generated application.

There follows a brief summary of each of the functions these supplementary utilities perform. A more detailed treatment can be found in the IDEAS User Manual.

- Check File Status - Provides important information on the status of each file, including the number of records specified and provided, % of file that is full, and number of overflow records.
- Reconstruct Key File - Reconstructs a key file in the event that the key file is damaged or destroyed, or it creates a new alternate.
- Convert IDEAS Data File to Telecommunications Format - The screen for this utility asks for the following: IDEAS file name, TC file name, output address for T/C file, number of sectors to be reserved for the output file, and if records are concatenated.
- Protect all Records in a File - Limits access to records in a file to one MVP or LVP partition.
- Release all Records in a File - Does the exact opposite of the "Protect All Records Utility." It allows a file to be accessed by any MVP or LVP partition.

## INQUIRY DATA ENTRY ACCESS SYSTEM

### SYSTEM RESIDENT MACROS

All application programs generated by IDEAS use the system resident MACROS. The MACROS are a set of 59 powerful subroutine calls which substantially reduce programming effort by automatically providing a number of frequently used file maintenance, data packing and unpacking, and disk operations routines. There are also routines for specialized operations such as Julian date conversion and setting a status flag. The MACROS for the 2200 MVP are contained in the program ID-SUB-M which is contained on the IDEAS Application Utilities Diskette. (Both the Supplementary Data File Utilities and System MACROS, then, are on the Application Utilities Diskette). A complete listing of these MACROS with DEFFN numbers is included in the Appendix.

**SUMMARY**

The single most appropriate word to describe IDEAS is "powerful." This word is overused--it is mentioned to describe a number of different computers and software packages--but in the case of IDEAS, the word "powerful" has meaning. IDEAS represents a new generation of software development tools. Previously, a programmer had to take time and effort to create screens, masks, data entry routines, and reports. IDEAS allows these activities to be done in a few operations at the screen, where feedback is immediate. In the past, revision and documentation were often difficult to accomplish. With IDEAS, revision and documentation are built into every utility. IDEAS is truly a powerful application development package because it accomplishes a number of tasks quickly and effectively: It generates its own code, and it can design, revise, and document with the same proficiency it performs complex operations based on input from a few, relatively simple screens. IDEAS, because it is such an unparalleled development tool, greatly enhances the capability of 2200 Series computers.

The following summary table lists each IDEAS utility, and the name and description of each program that comprises it. The individual program names and descriptions are only important in that they can provide an understanding of how each utility is structured. The table may also be useful as a reference if you wish to examine IDEAS more closely.

Table 2. Summary Table of IDEAS Utilities

IDEAS Utility	Program Name	Program Description
DATA FILE UTILITIES	<ol style="list-style-type: none"> <li>1. IDEAS31M</li> <li>2. IDEAS310</li> <li>3. IDEAS311</li> <li>4. IDEAS312</li> <li>5. IDEAS313</li> <li>6. IDEAS314</li> <li>7. IDEAS315</li> </ol>	Menu Create/revise/re-initialize module Record field definition module Key field selection module Performance option selection module File initialization module Documentation module
SCREEN MASK UTILITIES	<ol style="list-style-type: none"> <li>1. IDEAS32M</li> <li>2. IDEAS320</li> <li>3. IDEAS321</li> <li>4. IDEAS322</li> <li>5. IDEAS323</li> <li>6. IDEAS32X</li> </ol>	Menu Creation module Revision module Documentation module Printing module Execution
START PROGRAM GENERATOR	<ol style="list-style-type: none"> <li>1. IDEAS35M</li> <li>2. IDEAS350</li> <li>3. IDEAS351</li> <li>4. IDEAS352</li> </ol>	Menu Creation module Revision module Definition module
APPLICATION MENU PROGRAM UTILITIES	<ol style="list-style-type: none"> <li>1. IDEAS36M</li> <li>2. IDEAS360</li> <li>3. IDEAS361</li> <li>4. IDEAS362</li> <li>5. IDEAS363</li> </ol>	Menu Menu screen and program revision mod. Menu screen and program creation mod. Menu screen and program revision mod. Documentation program revision mod.
DATA ENTRY/ INQUIRY/ UPDATE PROGRAM GENERATION	<ol style="list-style-type: none"> <li>1. IDEAS37M</li> <li>2. IDEAS37X</li> <li>3. IDPROG01</li> <li>4. IDPROG02</li> <li>5. IDPROG03</li> <li>6. IDPROG04</li> <li>7. IDPROG05</li> <li>8. IDPROG06</li> <li>9. IDPROG07</li> <li>10. IDPROG08</li> </ol>	Program generation Module 1 of generated program Inquiry module Add module Add/modify module Add/delete module Add/modify/delete module Modify module Delete module Modify/delete module
REPORT/FORM PRINTING UTILITY	<ol style="list-style-type: none"> <li>1. IDEAS33M</li> <li>2. IDEAS330</li> <li>3. IDEAS331</li> <li>4. IDEAS332</li> <li>5. IDEAS333</li> <li>6. IDEAS334</li> <li>7. IDEAS33X</li> </ol>	Menu Creation/revision module Report generator module Execution module Execution module Documentation module Execution module



REVIEW QUESTIONS

1. Which of the following statements best represents the purpose of the IDEAS software package?
  - a. It allows the inexperienced computer operator to control both the input and output of 2200 Series computers.
  - b. It is a tool which simplifies and expedites programming and helps to create a framework around which applications can be built.
  - c. It is a system which by itself creates complete application programs and is intended for programmers with at least a moderate amount of data processing experience.
  - d. It is a series of subroutines (which can be entered from application programs) that perform a variety of file maintenance and record keeping functions.
2. Which 2200 Series computers can use IDEAS?
  - a. 2200 MVP only.
  - b. 2200T, VP, MVP and LVP only.
  - c. 2200T, VP, MVP, LVP and SVP.
  - d. 2200 MVP and LVP only.
3. Which IDEAS Utility should be used to design a special screen for data entry?
  - a. Data Entry, Inquiry, Update Program Generation.
  - b. Report/Form Printing.
  - c. Screen Mask.
  - d. Data File.
4. The IDEAS File Management System is called
  - a. KFAM-9.
  - b. FIMAN.
  - c. HIFAM.
  - d. HIKAM.

# INQUIRY DATA ENTRY ACCESS SYSTEM

---

5. Primary keys are stored in which file?
  - a. Data File.
  - b. Key File.
  - c. Alternate key file for sequential processing.
  - d. Alternate key file for random processing.
6. The final step in the creation of a data file is
  - a. Key Field Selection.
  - b. Data File Performance Option Selection.
  - c. File Name Specification.
  - d. Data File Initialization.
7. The following screen is from which IDEAS Utility?

```
I.D.E.A.S. System Utility - Create / Revise Screen "PSCRN"      Release 1.1
=====
'FM Field parameter -----
'00 Field name ? ..... "DESCRPTN" Valid Character List
'01 Row on screen ? ..... 1      1 Digits only
'02 Column on screen ? ..... 1    2 Digits & decimal pt
'03 Position in Record ? ..... 23 3 Digits & signs
'04 Default field ? ..... No default field 4 Digits, signs, & dec
'05 Field length ? ..... 25      5 Upper case letters
'06 Valid characters ? ..... Any character 6 UC alpha & digits
'07 Allow keyboard entry ? ..... Yes      7 UC, digits, & punct
'08 Allow display ? ..... Yes          8 Any character
'09 Required or optional ? ..... Optional field 9 FM Keys, EDIT & EXEC
'10 Full if present ? ..... Need not be filled
'11 Left or right justified ? ..... Left justified
'12 Zero or space fill ? ..... Not applicable
'13 Number of decimal places ? ..... Not applicable
'14 Termination Full/EXEC ? ... Terminate when full
'15 Save field parameters
-----
'25 Delete current field '31 Cancel to mask editor
'26 Insert current field  EDIT Display screen mask

Touch the FUNCTION KEY corresponding to the desired operation.
```

- a. START Program Generation.
  - b. Menu Program Utility.
  - c. Screen Mask Utility.
  - d. Report/Form Printing Utility.
8. In the Screen Mask Editor Screen, which of the following occurs?
- a. The screen mask is designed.
  - b. The screen mask file name is initially entered.
  - c. The screen mask parameters are defined.
  - d. The screen is proofread by a masked man.
9. Which of the following statements is true about the Data Entry/Inquiry/Update Program Generation Utility?
- a. It employs a number of different screens.
  - b. It creates 8 different data entry programs.
  - c. It creates one data entry program that is very versatile and powerful.
  - d. It only works with password protection.
10. The Supplementary Data File Utilities are on which of the following diskettes?
- a. System Utilities Diskette.
  - b. Data Utilities Diskette.
  - c. Application (Run-Time) Utilities Diskette.
  - d. IDEAS Supplementary Diskette.

## INQUIRY DATA ENTRY ACCESS SYSTEM

### ANSWERS

1. b. is correct. IDEAS is best used as a development tool to create a framework around which application programs can be built. (It is also a possibility, but almost all IDEAS programs are modified or adapted.)
2. b. 2200T, VP, MVP, and LVP computers can use IDEAS.
3. c. The Screen Mask Utilities.
4. d. HIKAM (Hashed Index Key File Access Method) is IDEAS File Management System.
5. a. Primary Keys are stored in the Data File. Separate files are created for alternate keys.
6. d. is the correct answer.
7. c. The screen is designed by input to the Screen Mask Editor.
8. a. The screen is designed by input to the Screen Mask Editor.
9. b. This utility supports 8 different data entry programs.
10. c.

APPENDIX

LIST OF SUBROUTINES AVAILABLE IN IDEAS SYSTEM SUBROUTINE MODULES

DEFFN'32(N\$)	Get field parameters and display screen.
DEFFN'33(N\$)	Get field parameters for screen N\$, do not display screen
DEFFN'34(F)	Allow input from keyboard into field number "F".
DEFFN'35(E\$)	Display E\$ as error message on line 24, sound audio alarm
DEFFN'36	Display all fields starting at current field # "F".
DEFFN'37(Q)	Display field # "Q".
DEFFN'38(N\$)	Get field parameters, display screen, keep current record
DEFFN'39(N\$,P)	Get limits of file "N\$" on device # "P".
DEFFN'40(N\$,E\$)	Load program "N\$", display loading message "E\$".
DEFFN'41(V,K4,TO)	GET and unpack record from file number "V".
DEFFN'42(V,P)	Pack record and PUT it in file number "V".
DEFFN'44(R,C,K\$,L)	Display "L" bytes of "K\$" at row "R", column "C".
DEFFN'45(Q)	Retrieve contents of field number "Q".
DEFFN'46(Q,K\$)	Put contents of "K\$" into field number "Q".
DEFFN'47(P,K\$,L)	Copy "L" bytes to/from "K\$" from/to record position "P".
DEFFN'48(C,K\$(1),L)	Copy "L" bytes of "K\$(1)" to print buffer at position "C".
DEFFN'49(L)	Print "L" bytes of the print buffer, reset buffer.
DEFFN'50	Reset print buffer.
DEFFN'51(Q,M)	Copy field # "Q" to print buffer at screen column + "M".
DEFFN'52(Q)	Unpack all field parameters for field number "Q".
DEFFN'53(E\$)	Display "E\$" on line 24 as message to operator.
DEFFN'54(P,K\$(1))	Internal routine used in date validation routines.
DEFFN'55(A,EO)	Round value "EO" to field spec. dec., put in field # "A".
DEFFN'56(K\$,Q)	Date validation/conversion subroutine.
DEFFN'57(L)	Convert Julian date.
DEFFN'58(V,TO,K\$,DO)	Key file access subroutine (insert/retrieve/delete keys).
DEFFN'59(V)	Set up merge array for "FIND 1ST" - file # "V".
DEFFN'60(V,K\$)	Set up merge array for "FIND 1ST" = "K\$" - file # "V".
DEFFN'61(V,K\$,TO)	Find lowest key = "K\$" in file "V".

INQUIRY DATA ENTRY ACCESS SYSTEM

DEFFN'62(V,U,TO)	Find next higher key in file "V".
DEFFN'63(M)	Display "M" bytes of "K\$" at current row and column.
DEFFN'65(A\$,E\$,G\$)	Display "A\$" and "E\$" on line 24 - Sound alarm if "G\$" = "!".
DEFFN'66(V,TO)	Find lowest key in file # "V".
DEFFN'67(V,TO)	Find first physical key in file # "V".
DEFFN'68(TO)	Get next physical key from last file inquired into.
DEFFN'79(R,C,K\$,L)	Display "L" bytes of "K\$" at row "R"+1, column "C"+1.
DEFFN'80(P,L)	Internal subroutines - redimension S\$(P)L.
DEFFN'81(Q)	Unpack basic field parameter for field number "Q".
DEFFN'82(N\$)	Set "V" = file number of file named "N\$" (or zero).
DEFFN'83(M,N)	Position cursor at row "M"+1, column "N"+1.
DEFFN'84(V)	Pack record in Z\$() to Z\$() for file # V.
DEFFN'85(V)	Unpack record in Z\$() to Z\$() for file # V.
DEFFN'86(V)	Build key index element, record in Z\$() for file # V.
DEFFN'87(V)	Build key for unpacked record in Z\$() for file # V.
DEFFN'88(V,K\$(1))	Adjust key in K\$(1) for ascending/descending order.
DEFFN'89(V,R\$(1),DO)	Internal subroutine - record protect check/update.
DEFFN'90(N\$,P)	Set device # "P" = 2 if N\$="IDEAS" or "ideas".
DEFFN'91(N,DO\$(N))	Set status flag N.
DEFFN'98	Error message trap - must be in user program if used.
DEFFN'99	Function key trap - must be in user program if used.
DEFFN'141(N\$,K\$,TO)	Same as DEFFN'41, but with file NAME.
DEFFN'142(N\$,P)	Same as DEFFN'42, but with file NAME.
DEFFN'155(A,EO)	Same as DEFFN'55, but do not display resultant field.
DEFFN'159(N\$)	Same as DEFFN'59, but with file NAME.
DEFFN'160(N\$,K\$)	Same as DEFFN'60, but with file NAME.
DEFFN'161(N\$,K\$,TO)	Same as DEFFN'61, but with file NAME.
DEFFN'162(N\$,U,TO)	Same as DEFFN'62, but with file NAME.
DEFFN'166(N\$,TO)	Same as DEFFN'66, but with file NAME.
DEFFN'167(N\$,TO)	Same as DEFFN'67, but with file NAME.

COURSE EVALUATION FORM

Course Title \_\_\_\_\_

Your Job Title \_\_\_\_\_

Length of Time in Present Position \_\_\_\_\_

Length of Time in Related Field \_\_\_\_\_

Please indicate the number (1-5) which best describes your opinions of the following aspects of the course. A rating of "1" indicates STRONG AGREEMENT; "5" indicates TOTAL DISAGREEMENT.

STYLE OF PRESENTATION

- \_\_\_\_\_ logically organized
- \_\_\_\_\_ well-written
- \_\_\_\_\_ understandable language
- \_\_\_\_\_ maintained interest
- \_\_\_\_\_ well-paced
- \_\_\_\_\_ appropriate and realistic examples or case studies

EXERCISES

- \_\_\_\_\_ sufficient quantity and frequency
- \_\_\_\_\_ well-integrated with course content
- \_\_\_\_\_ facilitated and reinforced learning
- \_\_\_\_\_ focussed on important areas
- \_\_\_\_\_ degree of difficulty appropriate for subject matter

MODULE TESTS

- \_\_\_\_\_ clearly written questions
- \_\_\_\_\_ tested material presented
- \_\_\_\_\_ appropriate degree of difficulty

AUDIO/VISUALS

- \_\_\_\_\_ sufficient number
- \_\_\_\_\_ high-quality
- \_\_\_\_\_ well-integrated with text
- \_\_\_\_\_ clarify important points made in text

OVERALL COURSE IMPRESSIONS

- \_\_\_\_\_ materials are job-related
- \_\_\_\_\_ technically accurate
- \_\_\_\_\_ material provides a logical flow from previous course(s) in this series (if applicable)



Please feel free to make additional comments, especially about any statement with which you indicated total disagreement.

-----  
fold here

Jim Walsh, Director  
Corporate Education Center  
Wang Laboratories, Inc.  
20 South Avenue  
Burlington, MA 01803

-----  
fold here





# **WANG EDUCATION CENTER**

**WANG**

**ONE INDUSTRIAL AVENUE, LOWELL, MA 01851**