TECHNICAL BULLETIN

DALLAS SMALL/ INTERMEDIATE SYSTEMS CENTER

Datamaster Procedures and Techniques

G360-1000-0
September 1984

International Business Machines Corporation

3820 State Street
P.O. Box 3467
Santa Barbara, California 93130
805/569-3400

December 20, 1984

Dear Datamaster User:

During the last few months, two new sources of S/23 technical information have been made available by the Dallas Small/Intermediate Systems Center. Some of you may have already received a copy of the Question and Answer Data Base diskette or the Procedures and Techniques technical bulletin.

The Question and Answer Data Base is a set of commonly asked questions, and the answers to those questions. It is available on diskette to all S/23 users, and comes with a program for accessing the information.

The Procedures and Techniques technical bulletin is a manual which contains a great deal of useful information about the Datamaster. Topics in the bulletin include problem determination, Customer Support Functions, BASIC programming techniques, and BRADS techniques.

I am enclosing a copy of the Procedures and Techniques manual in this package. I hope you will find it helpful. If you have not already received a copy of the Question and Answer Data Base, and would like a copy, please give me a call at (805)658-1111. Also, feel free to contact me if there is anything else I can do for you.

Sincerley,

Jill George

Jill M. George
Account Systems Engineer

IBM

TECHNICAL BULLETIN

DALLAS SMALL/ INTERMEDIATE SYSTEMS CENTER

Datamaster Procedures and Techniques

G360-1000-0
September 1984

THE INFORMATION CONTAINED IN THIS DOCUMENT HAS NOT BEEN SUBMITTED TO ANY FORMAL IBM TEST AND IS DISTRIBUTED ON AN 'AS IS' BASIS WITHOUT ANY WARRANTY EITHER EXPRESSED OR IMPLIED. THE USE OF THIS INFORMATION OR THE IMPLEMENTATION OF ANY OF THESE TECHNIQUES IS A CUSTOMER RESPONSIBILITY AND DEPENDS ON THE CUSTOMER'S OPERATIONAL ENVIRONMENT. WHILE EACH ITEM MAY HAVE BEEN REVIEWED BY IBM FOR ACCURACY IN A SPECIFIC SITUATION, THERE IS NO GUARANTEE THAT THE SAME OR SIMILAR RESULTS WILL BE OBTAINED ELSEWHERE. CUSTOMERS ATTEMPTING TO ADAPT THESE TECHNIQUES TO THEIR OWN ENVIRONMENTS DO SO AT THEIR OWN RISK.

REFERENCE TO FIXES (PTFs, APARs, ETC.) THAT HAVE NOT BEEN RELEASED THROUGH THE NORMAL DISTRIBUTION PROCESS DOES NOT IMPLY GENERAL AVAILABILITY. THE PURPOSE OF INCLUDING THESE REFERENCE NUMBERS IS TO ALERT IBM CUSTOMERS TO SPECIFIC INFORMATION RELATIVE TO THE IMPLEMENTATION OF THE FIX WHEN IT BECOMES AVAILABLE TO EACH CUSTOMER ACCORDING TO THE NORMAL IBM PTF DISTRIBUTION RULES.

REQUESTS FOR COPIES OF IBM PUBLICATIONS SHOULD BE MADE TO YOUR IBM REPRESENTATIVE OR TO THE IBM BRANCH OFFICE SERVING YOUR LOCALITY. ADDRESS COMMENTS CONCERNING THE CONTENT OF THIS PUBLICATION TO IBM Corporation, IBM Tower at Williams Square, Department 8RN, P.O. Box 160969, Irving, Texas 75016-0969.

IBM MAY USE OR DISTRIBUTE WHATEVER INFORMATION YOU SUPPLY IN ANY WAY IT BELIEVES APPROPRIATE WITHOUT INCURRING ANY OBLIGATION TO YOU.

CONTENTS

(This page intentionally left blank)

HOW TO ANALYZE AN ERROR

When an error occurs in processing, locate the error message number in either the Application's MESSAGES manual or the SYSTEM MESSAGES manual. If the error message does not seem to relate correctly to the action just taken or the line of code just executed, then it is a good idea to terminate the job you were attempting to run and return to READY INPUT. Now you can power down the system and when you power the system back ON do NOT enter the PROC START command. Instead, press the ERASE key and type 'PROC SETUP1', (refer to the section "Description of PROC SETUP1," page 15), and then press ENTER. When the CPU returns to READY INPUT, run your normal PROC START.

If the system did not completely load microcode correctly or has encountered a significant enough power variance, the Datamaster can display seemingly unrelated errors, traps, or go back through power ON diagnostics. Examples of the errors are 0726 and 4260, both of which may seem unrelated when they occur.

If the error seems to be related to the CSF microcode, VOL002, be sure you are operating at level 05, (refer to "How to Determine the CSF Level of the System," page 18; "How to Determine the Level of a Particular CSF File," page 19; "How to Update Your System to the Current CSF Level," page 19 for diskette and page 22 for 5247 Disk system).

If you encounter a TRAP or your job is interrupted by the Datamaster returning to Power ON diagnostics, refer to the section below, "What to do About TRAP Errors."

If you are having frequent occurrences of errors that indicate faulty diskettes, you should first analyze the environment before suspecting a diskette drive problem. The most frequent cause of a failing diskette is contaminants on the diskette surface, and the most common contaminant is finger prints. Other common contaminants are: dust particles, cigarette smoke particles, and human hair. Another frequent cause of diskette failure is corrupted sector addresses. This is caused by passing the diskette through an electro-magnetic force, such as that generated by the ring on a conventional telephone and many internal alarm systems that are located within two feet of the source.

WHAT TO DO ABOUT TRAP ERRORS

There are several possible causes for TRAP errors; of them, microcode problems, hardware malfunctions, and power fluctuations are the most likely. The appropriate steps to take depend on the type of TRAP error that occurs and whether or not it is reproducible.

In some cases a storage dump is recommended, but this cannot be done unless a storage dump diskette has been prepared. Each computer or work station should have its own storage dump diskette available. This must be a Type 1 diskette. Prepare it following the steps in the OPERATOR REFERENCE manual (SA34-0108) "Help" section item "Preparing a diskette to save list storage information" (use PROC STGDUMP). If this has not been done, do it at the earliest opportunity.

When a TRAP error occurs, first determine the most likely cause (see item "How to Interpret TRAP Errors," page 3).

If it appears to be hardware caused, turn off the computer, turn it back on, redo PROC START, close any open files (users should contact their programmer for the most appropriate recovery), and retry the operation that failed. If a hardware TRAP error occurs again, call IBM Service. Leave the TRAP on the computer until IBM Service arrives; it will help in diagnosing the problem.

If it happened as a result of a power fluctuation, turn off the computer, turn it back on, redo PROC START, close any open files (users should contact their programmer for the most appropriate recovery), and retry the operation that failed.

If it appears to be a microcode caused problem, take a storage dump: follow the steps in the OPERATOR REFERENCE manual (SA34-0108) "Help" section, item "What to do when you get a computer trap." Record all the hexadecimal information on the bottom lines of the display following the word TRAP and the steps taken leading up to the TRAP error. Turn off the computer, turn it back on, redo PROC START. At this point users should refer to page 3, concerning microcode related errors, as there are no reasons for a microcode TRAP when operating at the current level of CSF with the enhancements patches, (listed on page 31), applied correctly.

Programmers should check the items titled "A TRAP Error Listing a Program" and "A TRAP Error Running a Program" on page 37 to see if it is one of the known microcode caused TRAP errors. A listing of the Storage Dump can be of assistance in identifying the program statement executing when the TRAP error occurred (see item "How to Use the List Storage CSF," page 5).

If it is one of the known problems, do what is suggested to avoid the problem in the future. If it does not seem to be one of the known problems, have the user close the files, do whatever recovery is appropriate to the application and retry the failing operation.

HOW TO INTERPRET TRAP ERRORS

The following is an overview of many different types of TRAP errors and their most likely causes. This is not all inclusive, nor are these the only causes for TRAP errors.

If the TRAP error occurs during power-on diagnostics or the running of PROC START or PROC SETUP1, it is most likely a hardware problem or a faulty CSF diskette.

When a TRAP error occurs, one or two lines of hexadecimal information is usually displayed on the bottom of the display. The first two or four hexadecimal digits provide a general classification of the TRAP error. These can be used to determine the initial approach to resolving the problem (see item "What to do About TRAP Errors," page 2).

The following is a guideline for TRAP errors. It is not all inclusive, nor is it absolute.

TRAPS can be broken down into four main categories:

HARDWARE - This requires IBM Service to diagnose and repair.

FIRMWARE - This is caused by the operating system or Customer Support Functions. The correction for this is to get a new copy of the current CSF diskette, (EC 337334), and refer to the section "How to Update Your System to the Current CSF Level," on page 19 or page 22 for 5247 Disk. If this fails to correct the problem, then it is due to the part of the firmware hardwired into the system and will require IBM Service to repair.

SOFTWARE - This is a coding error and will require analyzing the program to correct. The use of STGDUMP and LISTSTOR will be most helpful.

INTERMITTENT - This is the most common type of customer reported TRAP and is generally a result of: variation in the building power*, failing diskette, environment, operator error, etc.

* A clean ground and a dedicated power line are essential in this area. Furthermore, if you are operating a system that includes a 5247 Disk Unit, it may be necessary to have

all 5322s, 5324s and the 5247 all attached to the same dedicated power supply and sharing a common ground. At first occurrence of a TRAP, power OFF the CPU and re-run PROC START and then CLOSE all open files. At this time PREPARE a storage dump diskette, (type 1 only!), and re-try the job that just resulted in the TRAP. If the same TRAP occurs, you can do a STGDUMP and then use LISTSTOR and the information listed below to analyze the problem.

TRAP # LIKELY CAUSE

0000 Usually caused by a partial load of feature microcode (Communications or Word Processing). This can be caused by a bad feature diskette. Follow the procedure for PROC SETUP1 and retry the job.

0004 Possible microcode problem. Make sure you are at level 05 CSF and follow the procedure for running PROC SETUP1 and then retry the job. If the trap continues, call IBM Service for a possible hardware problem.

0005 Bad copy of program in memory and/or on disk/ diskette. Try reloading the program, save as source, and load from the source copy.

0011 Usually caused by a bad diskette. Try copying the diskette (do not use IMAGE COPY) or restore from a backup.

10xx Memory parity error. This is a hardware caused TRAP error. Retry the operation after recovery. If it fails again, call IBM Service.

20xx I/O channel error, usually a thermal check. This is a hardware caused TRAP error. Retry the operation after recovery. If it fails again, call IBM Service.

40xx Probably one of the known microcode problems (see item "TRAP Error Running a Program," page 37). It can also be caused by a hardware malfunction. The trap is almost always caused by software; however, if the cause seems to be hardware, call IBM Service for assistance.

80xx Power check. This is a hardware caused TRAP error. Retry the operation after recovery. If it fails again, call IBM Service.

HOW TO USE THE LIST STORAGE CSF

The List Storage CSF (LISTSTOR) is a utility to print the contents of a storage dump diskette. Its primary purpose is to assist IBM in diagnosing TRAP errors caused by microcode problems. However, it can also be used as a diagnostic tool by a programmer.

Taking a storage dump requires the preparation of a storage dump diskette. This must be a Type 1 diskette. Prepare it following the steps in the OPERATOR REFERENCE manual (SA34-0108) "Help" section item "Preparing a diskette to save list storage information" (use PROC STGDUMP).

To take a storage dump:

1. Press the Hold key when the storage dump is needed (during execution of a program, at a pause, error, etc.)

2. Press the Test key and then the '9' key on the typewriter part of the keyboard (not the numeric pad)

3. Insert the prepared storage dump diskette into slot 1 (slot 3 if there are no slots in the CPU and a 5246 is attached). If dumping to the 5247 Disk Unit, no diskette is required. Go to the next step

4. If dumping to diskette, press the 'F' key and the '0' (zero) key on the typewriter keyboard. If dumping to the 5247 Disk Unit, press the 'F' key and the 'D' key

5. When the dump is completed (indicated by reverse image block on the display), press the 'E' key to restore the computer to the HOLD state set in step 1.

When the dump has been taken, it can be listed at anytime. To list a storage dump use the LISTSTOR CSF. Note, before listing a storage dump that was written to the 5247 Disk Unit, it must be copied to diskette. This diskette is prepared as described above. Use Option 5 on the List Storage Information Menu: "Copy Storage Information from Disk to Diskette" to transfer it to the prepared diskette.

The most useful form of storage listing is created by selecting Option 2 on the List Storage Information Menu: "List Storage Information using Recommendations." This prints a formatted listing including:

System Features and Status

BASIC Program Status (if a program is in memory)
- Line being executed
- Variables and their current values
- Procedures active

BASIC Program Open Files (if any)
- File name/VOLID/device
- How opened
- Last record accessed

More information about the LISTSTOR CSF can be found in the CUSTOMER SUPPORT FUNCTIONS VOLUME II manual, Book 6, (SA34-0176).

HOW TO USE THE LIST FILE CSF

When developing or modifying programs that use data files, some of the most difficult problems to diagnose are caused by incorrect data in files. The LISTFILE CSF can be used to print a status, full or partial listing of a data file. It prints the contents of the data file in both character and hexadecimal. Some of the options are:

- List the file sequentially
- List the file using an index file
- List a range of records or keys
- List deleted records
- List records beyond the end of data
- List summary data only
  - File type
  - Number of active records
  - Number of deleted records
  - Record length

This can also be useful in determining the format of records in a file. However, there are no field delimiters in internal files; therefore, it may be difficult to tell where one field ends and another begins.

More information about the LISTFILE CSF can be found in the CUSTOMER SUPPORT FUNCTIONS VOLUME II manual, Book 6, (SA34-0176).

ERROR 0102 RE-DIMENSIONING AN ARRAY

An invalid error 0102 may occur when re-dimensioning an array using a subscripted variable as a parameter of the re-dimension statement. For example:

```
00100 MAT A=A(X(2))
```

This can be avoided by first assigning the subscripted variable to an un-dimensioned variable:

```
00090 LET Y=X(2)
00100 MAT A=A(Y)
```

ERROR 0301 EXECUTING A USER FUNCTION

An invalid error 0301 may occur executing a user-defined function when the header line (DEF) of the function has been changed.

This can be corrected by saving the program as source and reloading it from the source file:

```
SAVE pgm.name.src/volid,SOURCE
LOAD pgm.name.src/volid
```

Then replacing it to the original file:

```
REPLACE pgm.name/volid
```

Also see the item titled "Program Editing Considerations" on page 35.

ERROR 0405 USING SRCH

Error 0405 occurs using SRCH that specifies a starting row number larger than 255.

This can be avoided by not searching arrays of more than 255 elements (see item "The System Function SRCH Works Improperly," page 39).

ERROR 0726 IN AN I/O STATEMENT

When an error 0726 occurs in an I/O statement with a large variable list or array variables, it can be difficult to determine which variable or element failed. The System Variable CNT is useful here. It contains the number of the

last I/O variable or array element processed successfully. To use it, first assign it to an ordinary variable, then print that variable. For example:

1. An error 0726 occurs at line 00200

2. Press the Error Reset key

3. Enter: LIST 200

4. The computer displays:

   00200 WRITE #1,USING 210:MAT A

5. Enter: XX=CNT

6. Enter: PRINT XX

   The computer displays:

   5

7. The error occurred trying to write the sixth element of the array A

8. Enter: PRINT A(6) and compare the result with the FORM specification on line 210.

ERRORS USING ARRAYS

If an error occurs that does not have an obvious cause like 0726 and the line that failed is using an entire array (such as, it contains a MAT parameter), check to insure that the correct OPTION BASE has been specified.

Most programmers use base 1, but the computer defaults to base 0. It is a good idea to code OPTION BASE 1 as the first line of each program unless base 0 is desired. In base 0 each array has one more element (the 0th element) than is specified in the DIM statement. For example, the array A with dimensions, DIM A(10), in base 0 has 11 elements, numbered: 0,1,2,3,4,5,6,7,8,9,10.

ERROR 0726 OR OTHER MAPPING ERRORS

Error 0726 or other mapping error may occur if the replication factor for a FORM statement specification is greater than 255. This is a design limit of the computer and is documented in the BASIC LANGUAGE REFERENCE manual (SA34-0109).

This problem can be avoided by 'breaking' the specifications into pieces smaller that 255. For example:

If the FORM statement wanted is:

```
00100 FORM 400*PD 5,C 10
```

Then use:

```
00100 FORM 200*PD 5,200*PD 5,C 10
```

ERROR 0802 USING PIC(Z.ZZ) FORMAT SPECIFICATION

Error 0802 occurs when running a program that has a FORM statement with a PIC specification with zero suppress Z's after the decimal point.

The Datamaster does not support this type of specification. This is documented in the BASIC LANGUAGE REFERENCE manual (SA34-0109).

ERROR 4000 DOING PROC START

Error 4000 occurs during PROC START for a file named IMFLMxx (where xx is the work station ID) if PROC SETUP1 has not been run for this computer or work station.

Turn the computer off, turn it back on, and enter PROC SETUP1 over the PROC START that the computer provides. See the item titled "Description of PROC SETUP1" on page 15.

ERROR 4000 OR 4159 DOING PROC SETUP1

If an error 4000 or 4159 occurs doing PROC SETUP1 it usually means that the diskette being used does not have a VOLID of VOL002.

This can be corrected by changing the VOLID to VOL002 using the VOLID command.

ERROR 4138 DURING INDEX

This error occurs when there is insufficient contiguous space starting at a track boundary available for the index file. Even though a DIR of the diskette shows sufficient space for the index file, it may not all be contiguous or start on a track boundary.

To prevent this problem from occurring, implement the procedure outlined in the topic titled "How Often Should INDEX be Run" on page 27.

ERROR 4152 AND 4000 DURING RESTORE

When using the BACKUP.RESTORE if you choose the option: "N - Remove Deleted Records," (screen #44-082), the backup will complete but the diskettes may be inaccessible when restoring, indicated by error 4000 or 4152. Therefore, NEVER use this option while in the BACKUP process. During BACKUP choose the option: "Y - Keep Records in Place."

The removal of deleted records can be accomplished during the RESTORE process without error.

4270 ERROR

The 4270 error has caused much confusion among users because of the many different reasons it occurs. It is often the result of a "Phantom File," (refer to the section, "Phantom File," page 13). Here is a listing of the causes of unexpected 4270 Errors and the circumventions for them.

1. The user is not operating at CSF level 05, (EC 337334).

Solution: Get a new CSF diskette, (refer to the section, "How to Acquire a New CSF Diskette," page 17), and follow the directions in section "How to Update Your System to the Current CSF Level," (page 19 for diskette systems, page 22 for 5247 Disk Unit systems).

2. The user has failed to run PROC START or has run a modified, damaged, or down level version of PROC START that has not properly LINKed UPDATE, which loads the microcode to prevent the "PHANTOM FILE."

Solution: If you now have the "PHANTOM FILE" then you must use BACKUP.RESTORE to backup all files possible, on all volumes. Then IBM Service will have to reformat the Disk. You can then follow the directions in "How to Install the Current CSF Level to Disk" and then use ADD.DELETE and BACKUP.RESTORE to restore files to the Disk.

Always run PROC START when a CPU is powered ON. And if you notice a power fluctuation while operating the system, power it OFF immediately, as microcode updates may have been lost. Then power it back ON and run PROC START so that the microcode updates can be reloaded to ensure proper functioning of the CPU, and then continue processing.

3. The user ran PROC SETUP3 and PROC SETUP1 from incompatible levels of VOL005 and VOL002 or ran them incorrectly. Refer to the section, "How to Update Your System to the Current CSF Level, for 5247 Disk Systems," page 22 for this procedure.

4. The user is running a program that uses the 'RELEASE' statement incorrectly. The Basic 'RELEASE' statement can only be used with a file opened with the 'SHRU' status. Using 'RELEASE' with any other file sharing status will result in unpredictable errors and can damage your data file, which may result in a 4270 Error.

Solution: Change the program containing the RELEASE statement to OPEN the file SHRU or change the logic so that the release statement can be deleted.

5. The data file has been damaged by any number of causes including item numbers 1-4 above, or such things as: a power surge, operating the disk in a room that is over 90 degrees Fahrenheit, hardware failure, etc.

Solution: Write a Basic program to rebuild the data file. This would involve a loop to read the file REC=x with an ON ERROR GOTO statement to by-pass any damaged records; otherwise, write the record to a new file. The form statement should read the data exactly as the form statements used in your normal programs. If the program loops on the 4270 error, then program should print out the record number and close the file. Re-open it reading REC=y, (where y is the number of the record just printed plus 2).

6. The system is executing a Procedure which contains 'LINK (CSF name)' but doesn't have an 'ENDLINK' statement, or has it spelled 'END LINK' (blank not allowed).

Solution: Correct the PROC file

7. The user is attempting to LOAD a program that has been damaged or was not saved correctly because a 4137 Error was encountered when the program was REPLACED.

Solution: Restore from a backup copy of the program or re-key the program

8. Attempting to read an INDEXED file that contains only deleted records or no records.

Solution: Use COPY or BACKUP.RESTORE to remove deleted records from the file and then enter records into the file and rebuild the INDEX

ERROR 6126 USING SUBSTRING IN INPUT FIELDS SPECIFICATION

Error 6126 may occur if the INPUT FIELDS specification contains a substringed variable. For example:

```
00100 INPUT FIELDS "2,"&COL$(2:3)&",C 8,N,N":X$
```

This error can be avoided by assigning the substringed variable to another variable and using that variable in the INPUT FIELDS specification. For example:

```
00090 LET CC$=COL$(2:3)
00100 INPUT FIELDS "2,"&CC$&",C 8,N,N":X$
```

ERROR 6401 AND 6480

These errors are almost always a result of having run PROC SETUP3 and PROC SETUP1 from incompatible CSF levels or from a system buffer being exceeded by a program looping on a continual error. The solution is to power OFF the system and then follow the directions on page 22, "How to Update your System to the Current level CSF for 5247 Disk."

If this fails to correct the error then you must have IBM Service check the system for a hardware problem. It may be necessary for IBM Service to reformat the Disk. If it is, use BACKUP.RESTORE to backup all volumes from the 5247 Disk Unit and then follow the directions in the section "How to Update Your System to the Current CSF Level for 5247 Disk," page 22.

ERROR 7501 OR 7503 DOING PROC SETUP1

Error 7501 or 7503 may occur doing PROC SETUP1 if PROC START has already been run, or a previous PROC SETUP1 has been run since the computer was turned on.

PROC SETUP1 must be done before PROC START and can only be run once after the computer is turned on. To avoid this error, turn the computer off, turn it back on and enter PROC SETUP1 over the PROC START provided by the computer. See the item titled "Description of PROC SETUP1" on page 15.

NO ERROR MESSAGE AND A SPLIT SCREEN

If an error occurs in a program that is being run as the last step in a procedure file, and PROCERR RETURN is in effect, the computer displays a 'Split Screen' without any error message.

To avoid this include a PROCERR STOP command in the procedure or trap and handle the errors in the program.

The error that occurred can be displayed by entering: PRINT ERR .

PHANTOM FILE

This term refers to a file that does not appear when running a DIR, but cannot be created because the system generates Error 4150 or a related error. It is also a file that appears on the DIR listing but cannot be removed using the FREE command and generates Error 4152 or a related error. A Phantom File may cause the computer to generate a number of different error codes, including: 601, 602, 4000, 4004-4007, 4055, 4148, 4159.

The solution to a "PHANTOM FILE" is to call IBM Service and have the 5247 Disk Unit reformatted. Failing to reformat the Disk will lead to additional damaged files.

Before IBM Service arrives you must use BACKUP.RESTORE to make backup copies of all the volumes on the 5247 Disk Unit, as reformatting will initialize the Disk just as PREPARE initializes diskettes.

The volume which contains the "PHANTOM FILE" may not backup because of the damaged file. If not, you will need to create a new volume on the 5247 Disk using LINK ADD.DELETE and then use LINK COPY, choosing option 2, and copy file by file from the damaged volume to the newly-created volume. Only good files will copy; therefore, you will be able to use LINK BACKUP.RESTORE to backup the new volume. Once IBM Service has reformatted the 5247 Disk, be sure to follow the steps in the section: "How to Update Your System to the Current CSF Level, for 5247 Disk Systems;" page 22, before using ADD.DELETE and BACKUP.RESTORE to restore all volumes.

FALSE ERROR MESSAGE FROM PROCEDURE

When running procedures that use PROCERR RETURN to suppress errors or reset the System Variable ERR to 0, it is possible to get an error message displayed that is not the true cause of the problem.

Some errors are not trapped by PROCERR RETURN (for example, 1008, 2104, 4000 and 4001). Therefore, if an error occurs that is trapped by PROCERR RETURN and is not handled properly by the procedure (by analyzing ERR), the procedure continues until an error which cannot be suppressed occurs. This masks the true cause of the problem.

To avoid false error messages be sure to reset the return with a PROCERR STOP command as soon as the steps for which errors are being trapped are completed. Don't use a 'blanket' PROCERR RETURN in procedures.

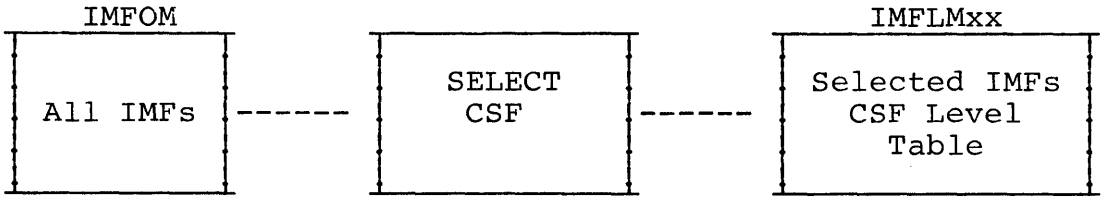
Also note that any PROCERR command issued in a procedure remains in effect for all subsequent procedures and sub-procedures, whether chained to a program or invoked from a procedure. Therefore, it is a good idea to begin any procedure with PROCERR STOP.

In order to reset the System Variable ERR to zero without suppressing errors include a PROCERR RETURN followed by a PROCERR STOP in the procedure.

DESCRIPTION OF PROC SETUP1

The SETUP1 procedure creates a file on the Customer Support Functions diskette (VOL002) or on the CSF volume (VOL002) on the 5247 Disk Unit. This file contains the machine updates appropriate for the computer or work station on which it is run. These updates are loaded into the update area of the computer each time PROC START is run.

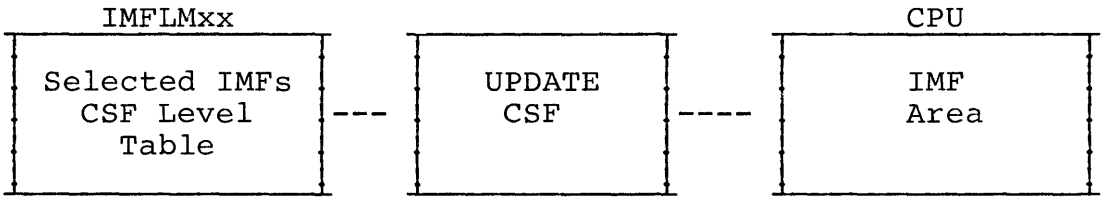
PROC SETUP1



where: xx is the work station ID (WSID$).

: CSF level table is the level numbers of the CSFs on the CSF volume used for the PROC SETUP1.

PROC START



PROC SETUP1 should only be run immediately after the computer is turned on, before PROC START is run. If it is run after PROC START or a prior PROC SETUP1, an error 7501 or 7503 may occur.

PROC SETUP1 must be run:

1. When the computer is first set up

2. When a feature (like Word Processing) or a device (like the 5247 Disk Unit) is installed or removed

3. When a new VOL002 diskette is received from IBM and the instructions require it

4. When a machine update received from IBM is applied and the instructions require it.

If the 5247 Disk Unit is installed, PROC SETUP1 must be run from each work station. Do not have any diskettes inserted when running PROC SETUP1.

If a work station is reconnected to a different port on the 5247 Disk Unit, PROC SETUP1 must be rerun for that work station.

DESCRIPTION OF PROC SETUP2

This procedure should only be run when first installing a computer or work station with more than one diskette drive. Insert the CSF VOL002 diskette into each additional drive and run PROC SETUP2. It verifies that the drive is attached properly. It does not do any diagnostics, change the computer or CSF volume (which PROC SETUP1 does).

DESCRIPTION OF PROC SETUP3

PROC SETUP3 reloads the 5247 Disk Unit I/O control programs onto the disk. It does not erase the disk or alter any user files, file headers, or volume labels, unless the Disk has encountered errors in the system area or is damaged. However, certain error conditions could cause PROC SETUP3 to initialize all systems pointers to the disk volumes which would make them inaccessible. Therefore, before you run PROC SETUP3 you need to have a current backup of your 5247 disk.

NOTE: You should have a procedure in process that makes a regular backup of your system for protection from data loss, just as you should always keep a backup copy of all diskette-based data to protect against diskette failure.

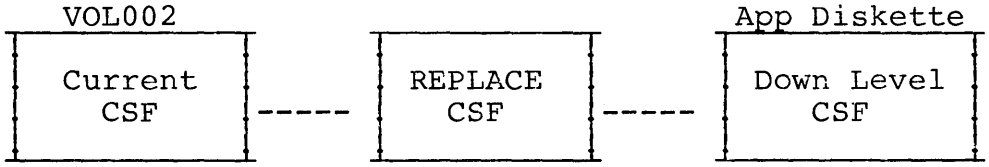
PROC SETUP3 should be run:

1 When a new VOL005 CSF diskette is received from IBM and the instructions require it

2. When planning to update your system to a new CSF level. The 5247 Disk Unit and all attached Datamasters must be powered OFF after PROC SETUP3 has completed and prior to running PROC SETUP1, refer to the section "How to Update Your System to the Current CSF Level, for 5247 Disk Systems," page 22.

DESCRIPTION OF REPLACE

The REPLACE CSF replaces down-level copies of the CSF routines on working copies of the CSF diskette(s), or those on application diskettes. It only replaces a CSF with a lower-level number (see item "How to Determine the Level of a Particular CSF File," page 19).



The REPLACE CSF should be run when a user receives a new CSF diskette from IBM and the instructions require it. If an application diskette does not have any CSFs on it or they are at the same or higher level, REPLACE does not replace them. The status display lists the CSFs as they are replaced.

If a 7010 Error persists after completing the Link REPLACE operation to that Customer Support Function, then the Customer Support Function is damaged. This can then be corrected by using Link COPY, instead of Link REPLACE. The COPY will replace the damaged Customer Support Function regardless of the level.

HOW TO ACQUIRE A NEW CSF DISKETTE

Customers who purchased their Datamaster directly from IBM can call the Datamaster Customer Assistance Center (CAC), 1-800-241-7501, and order the level 05, VOL002 diskette, (EC 337334). If the customer has a 5247 Disk Unit attached to the Datamaster, he needs to inform the Datamaster CAC of this attachment. The 5247 Disk Unit requires an additional VOL005 diskette, (EC 337334).

Those customers who purchased their Datamaster from an alternate source, such as a Value Added Remarketer, VAR, will need to contact that source for the new CSF diskette. The Datamaster CAC mails new CSF diskettes to VARs upon their request.

HOW TO DETERMINE THE HARDWARE LEVEL OF THE SYSTEM

When the Datamaster is first powered ON, before PROC START is run, the hardware level will be displayed on the right hand portion of the status line. The valid levels are: 1.01, 1.03, 1.04, 1.05. If any other numbers are displayed, such as: 1.02 or 2.5A, you should call IBM Service to correct the problem.

HOW TO DETERMINE THE CSF LEVEL OF THE SYSTEM

After the Datamaster has been powered ON and a PROC START has been completed, the level of CSF may be determined using the following procedure:

1. Press the HOLD key
2. Press the TEST key
3. Press the '9' key, (above the i and o keys only)
4. Press the 'D' key
5. Press '09806', (here again, use only the numeric keys found above the q - p character keys, not the numeric keys on the right of the keyboard.)

The screen should now be filled with hexadecimal data. The first row will show the system level and will appear as below:

```
9806   05 05 05 05 05 05 05 05 05 05 05 05 05 05 00 00
```

The '05' indicates that the system is at level 05 CSF, EC 337334. If '04' is displayed, then the system is at level 04 CSF. If the levels displayed are not all the same, the system is running at more than one level and you should follow the directions in the section "How to Update Your System to the Current CSF Level," page 19 or page 22.

To end this procedure, press the 'E' key and the screen will clear and display 'HOLD' on the status line. Now press the HOLD key and you will return to whatever screen you were on when you began this test. This test can be run from READY INPUT or in the middle of any procedure or program without interrupting the job.

THE EC NUMBER ASSOCIATED WITH VARIOUS CSF LEVELS

There are currently five levels of CSFs in use:

LEVEL	EC NUMBER OF VOL002	EC NUMBER OF VOL005
02	EC 466861	N/A
03	EC 997208	EC 997206
04	EC 998319	EC 335414
04	EC 336619	EC 335414
05*	EC 337334*	EC 337334*

* All 5247 Disk systems must be operating at level 05 and must have both the VOL002 and the VOL005 diskettes with this EC number.

NOTE: All hardware levels are compatible with all the above levels of CSFs; however, an IMF card is required for only the level 1.01 planer board.

HOW TO DETERMINE THE LEVEL OF A PARTICULAR CSF FILE

There are several different levels of Customer Support Functions. Many programmers have copies of selected CSFs on their applications diskettes. If the computer is set up with Customer Support Functions of one level and an application links its own copy of a CSF at a different level, an Option 95 Error 7010 occurs. To determine the level of any Customer Support Function, (any type 10 file), do the following:

1. Insert the CSF diskette. If using the 5247 Disk Unit, no diskette is required. Go to step 2

2. Type: LINK SELECT and press the ENTER key

3. Select Option 'P' on the menu. This option is not shown on the menu but is still valid

4. Select '1' for screen only

5. Enter the name of the CSF, (or any type 10 file), to be checked

6. Insert the diskette in question, (this can be a CSF diskette or other application diskette)

7. Enter the slot number containing the CSF to be checked. If it is on the 5247 Disk Unit, enter the volume id (usually VOL002)

8. When prompted, enter the offset of '00009'

9. The first two characters of "Original data" displayed are the Level of that CSF (for example, 03)

10. Use CMD 9 to return to the SELECT menu.

If the level of the CSF causing the error 7010 is less than the computer CSF level, the REPLACE CSF can be used to replace the application copy with the later computer version. If the level of the CSF causing the error 7010 is greater than the the computer CSF level, use the COPY CSF to copy the computer level to the application diskette.

HOW TO UPDATE YOUR SYSTEM TO THE CURRENT CSF LEVEL

For 5322 and 5324 Diskette System Only

To provide for correct update of your Datamaster to the current Customer Support Function level (EC 337334), please follow the outlined steps below. If your Datamaster is

attached to a 5247 Disk Unit, do not use this procedure. Instead refer to the section for 5247 Disk Systems below.

1. Power OFF your Datamaster

2. Power ON your Datamaster

3. Insert the new VOL002, EC 337334 into drive 1 and close the drive handle

4. Erase PROC START when it appears on the screen and type in PROC SETUP1, then press the ENTER key. When READY INPUT appears on the status line, type in PROC START and press the ENTER key

5. When READY INPUT appears on the status line, type in LINK COPY and press the ENTER key

6. When screen 03-010 appears choose option '2' to copy all files

7. On screen 03-082, accept the defaults by pressing the ENTER key.

8. On screen 03-092, answer 'Y' to the item: "Replace information in the output file." Accept the default answers for the remaining items by pressing the ENTER key

9. Now insert one of your old VOL002 Customer Support Functions diskettes into drive 2 and close the drive handle

10. Screen 03-102 should now be displayed. Fill out the screen as follows:

Input drive number	- 1
Input VOLID	- VOL002
Input filename	- *
Output drive number	- 2
Output VOLID	- VOL002
Output filename	- (leave this field blank)

After filling out screen 03-102 as above, press the ENTER key to begin the COPY operation. After the COPY has completed, follow the instructions on the screen to return to the COPY menu, screen 03-010. Remove the Output diskette from drive 2 and with a felt tip pen write EC 337334 on the label

11. Repeat steps 6 through 10, of this page, for all old VOL002, Customer Support Functions diskettes.

You have now completed updating all of your Customer Support Function diskettes to EC 337334. If you have more than one Datamaster, you need to run PROC SETUP1 as described in steps 1-4, page 19, using the Customer Support Function diskette for that work station.

You will now need to gather all program and data diskettes for your Datamaster. This includes working and backup copies of your program and data diskettes. The following steps should be taken to update all diskettes.

1. Insert the VOL002, EC 337334 diskette into drive 1 and type in LINK REPLACE and then press the ENTER key

2. On screen 11-010, choose option 2 to replace Customer Support Functions. When screen 11-042 is displayed, press the ENTER key

3. Screen 11-052 should now be displayed. Fill out the screen as follows:

Input drive number	- 1
Input VOLID	- VOL002
Output drive number	- 2
Output VOLID	- (leave this blank)

   After filling out screen 11-052 as above, press the ENTER key to begin the REPLACE operation

4. Follow the instructions on the screens to return to screen 11-010. Then repeat steps 13 and 14 above to replace the Customer Support Functions on all diskettes. You have now completed updating your Datamaster and all diskettes.

NOTE: Should you encounter an Error 7010 at anytime during processing the following procedure is recommended.

Make a note of the Customer Support Function you are attempting to use and the diskette on which it resides. Then press Error Reset to continue processing. When you have finished processing and returned to READY INPUT you need to use LINK COPY to correct the Customer Support Function that caused the error. To do this follow these steps:

1. With READY INPUT appearing on the status line, insert the Customer Support Function Diskette, VOL002, EC 337334 into drive 1

2. Type in LINK COPY and press the ENTER key

3. When screen 03-010 appears, choose option 2 to copy one file

4. On screen 03-082, accept the defaults by pressing the ENTER key

5. On screen 03-092, answer 'Y' to the item: "Replace Information in Output File." Accept the default answers for the remaining items by pressing the ENTER key

6. Now insert the diskette on which the error 7010 occurred into drive 2, and close the drive handle

7. Screen 03-102 should now be displayed. Fill out the screen as follows:

Input drive number	- 1
Input VOLID	- VOL002
Input filename	- (name of the Customer Support Function)
Output drive number	- 2
Output VOLID	- (leave these two
Output filename	- fields blank)

After filling out screen 03-102 as above, press the ENTER key to begin the COPY operation. After the copy has completed, follow the instructions on the screen to return to READY INPUT.

HOW TO UPDATE YOUR SYSTEM TO THE CURRENT CSF LEVEL

For a 5247 DISK SYSTEM ONLY

Follow the instructions below to install the current VOL002 and VOL005 Customer Support Functions on your system:

1. If your Datamasters and 5247 Disk Unit are powered OFF, then turn them ON. If your system is ON, go to step 3

2. Do your normal PROC START that you do each time the Datamaster is powered ON

3. Make sure that all the Datamasters attached to the 5247 Disk Unit are at READY INPUT

4. Insert VOL005, EC 337334 into drive 1 of one of the Datamasters that is attached to the 5247 Disk Unit. Then type PROC SETUP3 and press the ENTER key. Wait for READY INPUT to appear on the status line of the screen before removing the VOL005 diskette from the drive. Then place the VOL005 diskette in the blue IBM binder labeled 5247 MAINTENANCE DOCUMENTATION MATERIAL BOOK THREE. If you have any down level, (different EC number), VOL005 diskette you should use COPY to copy the new EC 337334 onto the old diskettes

5. Type in LINK COPY and press the ENTER key. From screen 03-010, choose option 2 to copy all files. On the following screen, 03-082, accept the defaults by pressing the ENTER key. On screen 03-092, answer 'Y' to the item: "Replace information in output file." Accept the default answers for the remaining items by pressing the ENTER key. At this point insert the VOL002, EC 337334, Customer Support Functions diskette into drive 1 and close the drive handle

NOTE: If you are now performing step 17 to update diskettes, then do not insert the VOL002 diskette into drive 1. Instead, you want to insert a diskette identified in step 16 for updating. When the update has been completed for each diskette, repeat the procedure for the next diskette.

6. Screen 03-102 should now be displayed on the screen. Fill out the screen as follows:

```
Input Drive Number  - 1
Input VOLID         - VOL002
Input Filename      - *
Output Drive Number - 5
Output VOLID        - VOL002
Output Filename     -     (leave this field blank)
```

After filling out screen 03-102 as above, press the ENTER key to begin the COPY operation. After COPY has completed, follow the instructions on the screen to return to READY INPUT

7. Remove the VOL002 diskette. Power OFF all Datamasters and the 5247 Disk Unit. Wait for the red power light on the 5247 Disk Unit to go OFF, and then turn it back ON. When the green ready light is ON, power ON the Datamasters

8. Erase PROC START when it appears on the screen and type in PROC SETUP1//5, then press the ENTER key. When READY INPUT appears on the status line of the screen, type in PROC START and press the ENTER key

REPEAT STEP 8 FOR EACH DATAMASTER ATTACHED TO THE 5247 DISK UNIT

9. Upon returning to READY INPUT, after the final PROC START has been completed, check to ensure that there are not any diskettes in the drives

10. Type in VOLID and press the ENTER key. This will produce a listing on the screen of the VOLIDs for the system. Use CMD/CopyD to obtain a printed copy of all the volumes on the 5247 Disk Unit

11. Insert the VOL002, EC 337334 diskette, into drive 1 and type in LINK REPLACE and then press the ENTER key

12. On screen 11-010, choose option 2 to replace Customer Support Functions. When screen 11-042 is displayed, press the ENTER key

13. Screen 11-052 should now be displayed on the screen. Fill out the screen as follows:

```
Input Drive Number  - 1
Input VOLID         - VOL002
Output Drive Number - 5
Output VOLID        - (the name of a volume)
```

    After filling out screen 11-052 as above, press the ENTER key to begin the REPLACE operation

14. Follow the instructions on the screens to return to screen 11-010. Then repeat steps 12 and 13 to replace the Customer Support Functions on all 5247 Disk Unit volumes whose VOLID is found on the VOLID listing from step 10 on page 23

15. Follow the instructions on the screen to return to READY INPUT. If your system has a Datamaster with two diskette drives, then go to step 18

16. Since your Datamaster has only one diskette drive, you cannot use LINK REPLACE to update the Customer Support Functions on your diskettes. Therefore, you must print a DIR listing of all your diskettes to determine which, if any, have Customer Support Functions on them. Customer Support Functions can be easily identified as they are all type 10 files

17. You must use LINK COPY to update the Customer Support Functions that appear on the DIR listings. These will have to be copied one file at a time using steps 5 and 6 with the following changes to screen 03-102:

```
Input Drive Number  - 5
Input VOLID         - VOL002
Input Filename      - (name of a CSF file)
Output Drive Number - 1
Output VOLID        -    (leave this field blank)
Output Filename     -    (leave this field blank)
```

    After completing this for all Customer Support Functions appearing on the DIR listings, go to step 19

18. A LINK REPLACE should be run on all diskettes. Follow steps 11 thru 14 on page 26 changing Output Drive Number in step 13 on screen 11-052 from '5' to '2.' Also, the volume name for Output VOLID will be the VOLID of the individual diskette that you have inserted into drive number 2

19. You have now completed updating your Datamaster and 5247 Disk Unit, and all diskettes.

NOTE: Should you encounter an Error 7010 at anytime during processing, use the following procedure:

Make a note of the Customer Support Function you are attempting to use and the diskette or Disk volume on which it resides. Then press Error Reset to continue processing. When you have finished processing and returned to READY INPUT, you can follow the instructions in step 17 on page 24 to update the Customer Support Function that is causing the error.

HOW TO COMPRESS A DISKETTE

When a diskette has files added and removed frequently, the diskette becomes fragmented. This reduces the amount of contiguous space available for new files (especially index files). The "Copy all files" option of the COPY CSF compresses the files together on the new diskette and consolidates all the free space into one area. It also consolidates all the files into one extent, which improves performance. It can also remove deleted records and reduce the space allocated for a file to only that which is required to hold the current records.

If deleted records are removed from an indexed file, the index file must be rebuilt using the INDEX CSF.

HOW TO COMPRESS A 5247 DISK UNIT

The 5247 Disk Unit does not store information by volume or by file; instead, it stores information sequentially. It is, therefore, necessary under heavy usage to compress the Disk Unit when you receive errors that indicate lack of Disk space or too many file extents. The procedure to use is as follows:

1. From a READY INPUT screen, with all drives empty, type in 'VOLID' and press ENTER. A list of the volumes on the 5247 Disk Unit will be displayed on the screen and should be printed at this time by pressing the 'CMD' key and the 'CopyD' key

2. LINK BACKUP.RESTORE to backup all volumes on the 5247 Disk Unit. Follow the instructions on the screen for backing up volumes. Be sure to choose the option to "Include Deleted Records", as removing deleted records during backup can render the files unusable

3. LINK ADD.DELETE to delete the volumes from the 5247 Disk Unit. Choose option 'H', (this option is not displayed on the main menu, but is valid), to delete a volume that contains files. Follow the instructions on the screen to delete all volumes except VOL002

NOTE: Remember option 'H' will delete an entire volume containing files and there is no recovery other than restoring from diskettes.

4. LINK ADD.DELETE to add the volumes to the 5247 Disk Unit. Follow the instructions on the screen throughout this step to add the VOLIDs on the list made in step 1 to the 5247 Disk Unit

NOTE: Remember that VOL002 must be the oldest volume on the Disk. Therefore, it must be the first volume ID specified when using ADD.DELETE. This will make it appear first on the list when you use the VOLID command to list the volumes on the 5247 Disk Unit. Failure to have VOL002 as the oldest volume on the Disk will result in unpredictable system operations and random errors.

5. LINK BACKUP.RESTORE to restore the volumes to the 5247 Disk Unit. Follow the instructions on the screen for restoring volumes. If your applications keep track of deleted record counts as IBM Business Accounting Management System (BMAS) applications do, then you must choose the option to "Include Delete Records;" otherwise you may choose the option to "Remove Deleted Records."

NOTE: IBM BMAS applications provide a menu option, (Reorganize), that is used to remove deleted records within the application so that the record counts are maintained.

SORT AND INDEX LIMIT

The SORT and INDEX CSFs only work on files containing less than 65535 records. This limit does not apply to reading files sequentially or directly. It is especially applicable to users of the 5247 Disk Unit.

HOW OFTEN SHOULD INDEX BE RUN

Having key records in the index file overflow area slows down access to the file. When records are added to an existing indexed file, the key records are put in an overflow area of the index file. These additional keys are not in sorted order. This causes the application to run more slowly.

Having many deleted records in an indexed file also slows down an application, because extra file accesses are needed to find the desired record. To avoid these problems the master file should be reorganized and a new index file created for it when significant additions or deletions occur. A procedure to do this could be a part of the regular backup procedure:

Copy the files diskette to the backup diskette, omitting deleted records

Copy (or sort) the master file(s) back to the files diskette

Run the INDEX CSF for each indexed master file.

NOTE: This procedure should not be used if using IBM BMAS applications, as they keep record counts in the control files which include deleted records. Therefore, when using IBM BMAS applications on diskette-based systems, use the job to "Copy/Rebuild" and on a 5247 Disk Unit-based system, use the job to "Reorganize."

HOW TO USE THE PRINTER SWITCH

The Printer Switch Feature allows two computers to be attached to the same printer. If both computers do not require the printer at the same time, it can be switched to the computer that needs it. However, the printer should not be switched to the other computer until the computer that is using it completes the application.

The following sequence must be followed when initially setting up the systems using the printer switch:

First Computer

1. Set the printer switch to the first computer
2. Turn on the 5246 and/or 5247 (if attached)
3. Turn on the printer

4. Turn on the first computer

5. Insert the CSF VOL002 diskette (if not attached to a 5247 Disk Unit) and type PROC SETUP1 over the PROC START provided by the computer and press ENTER

6. When READY INPUT is displayed, turn off the printer

Second Computer

7. Set the printer switch to the second computer

8. Turn on the printer

9. Turn on the second computer

10. Insert its CSF VOL002 diskette (if not attached to a 5247 Disk Unit) and type PROC SETUP1 over the PROC START provided by the computer and press ENTER

11. When READY INPUT is displayed, both computers are ready for operation.

Each time the computers are turned on, use the following sequence:

First Computer

1. Set the printer switch to the first computer

2. Turn on the 5246 and/or 5247 (if attached)

3. Turn on the printer

4. Turn on the first computer

5. Insert its CSF VOL002 diskette (if not attached to a 5247 Disk Unit) and press ENTER when PROC START is provided by the computer

6. When READY INPUT is displayed, turn off the printer

Second Computer

7. Set the printer switch to the second computer

8. Turn on the printer

9. Turn on the second computer

10. Insert its CSF VOL002 diskette (if not attached to a 5247 Disk Unit) and press ENTER when PROC START is provided by the computer

11. When READY INPUT is displayed, both computers are ready for operation.

When switching the printer from one computer to the other:

1. Turn off the printer

2. Set the printer switch to the other computer

3. Turn on the printer.

NOTE: If the printer mode has been changed, switching it to the other computer causes it to return to the default mode (10 cpi, draft, 6 lines/inch) because it was turned off for switching.

(This page intentionally left blank.)

HOW TO APPLY A MICROCODE PATCH

MAKE SURE YOU HAVE A BACKUP COPY OF THE CSF YOU ARE ATTEMPTING TO PATCH BEFORE CONTINUING

1. Using CSF VOL002 enter 'LINK SELECT'

2. On screen 10-010 enter a 'P' (for patch)

3. On screen 10-34P enter a '1'

4. On screen 10-26P enter the name of the CSF to be patched, (example: BACKUP.RESTORE)

5. On screen 10-27P enter the drive number and VOLID. (If patch to 5247 Disk Unit use drive # 5 and VOLID of VOL002. If patch to diskette, insert it in drive #1 and use VOLID of VOL002)

6. On screen 10-28P enter the five-character OFFSET, (from the patch you are applying)

7. On screen 10-29P enter the REPLACEMENT DATA and CRC, (from the patch you are applying)

NOTE: Check the original data that appears on each screen to verify that it is the same as the original data listed by the offset on the patch. If the data is not the same you either have down-level CSF diskette or a damaged CSF file. In either case, you cannot apply the patch to this CSF file.

8. On screen 10-28P enter 'END' for the OFFSET, then FIELD EXIT and ENTER to end the patch.

At this point you will see:

PATCH CUSTOMER SUPPORT FUNCTION SUCCESSFUL

If for any reason you are unable to COMPLETE THE ENTIRE PATCH, copy the CSF file from the unpatched backup copy over the partially patched version. Use of a CSF file that has only part of a patch applied will result in unpredictable ERRORS and TRAPS.

PATCHES TO LEVEL 05 CUSTOMER SUPPORT FUNCTIONS

The following pages contain five patches to level 05 of the Customer Support Functions. The application of these patches is recommended for the improved operation of the

CSF. These patches can only be applied to level 05, (EC 337334). If your system includes a 5247 Disk Unit, then it is NECESSARY for you to be operating at level 05 CSF. If you are not currently at CSF level 05 and are encountering CSF related ERRORS or TRAPS, you should obtain a level 05 CSF diskette, (refer to section, "How to Acquire a New CSF Diskette," page 17).

PATCH BR00252 FOR BACKUP.RESTORE

OFFSET	ORIGINAL DATA	REPLACEMENT DATA	CRC
06D50	2196B072	CDC69C72	5C91
000C6	00000000	D52A63B8	DFFC
000CA	00000000	EB010002	1FC0
000CE	00000000	21CE8CCD	35BB
000D2	00000000	339D2100	A57A
000D6	00000000	00CD94A8	9F81
000DA	00000000	CADE9C03	304B
000DE	00000000	D1C5E1CD	66CF
000E2	00000000	94A8D2E9	FF6E
000E6	00000000	9CE5D121	B74D
000EA	00000000	96B0C900	4408

END

PATCH BR00264 TO SORT

OFFSET	ORIGINAL DATA	REPLACEMENT DATA	CRC
00656	1F772B70	CD6A9C70	A9B6
0006A	00000000	1F772B7E	22D4
0006E	00000000	B1C00680	7ED6
00072	00000000	C9000000	65D8

END

PATCH BR00267 TO INDEX

OFFSET	ORIGINAL DATA	REPLACEMENT DATA	CRC
04FB4	3AF6A7FE	2AF6A77C	C83C
04FB8	01DABFB3	B5CABFB3	1E6F

END

PATCH BR00272 TO SORT

OFFSET	ORIGINAL DATA	REPLACEMENT DATA	CRC
027D6	3A97A6FE	2A97A67C	264B
027DA	01DAE1C3	B5CAE1C3	374D

END

PATCH BR00266 FOR ERROR 6573 IS APPLIED TO VOL005

This patch is applied with the same procedure as other patches; however, it is applied to VOL005 and not VOL002. Use VOL005 diskette, EC 337334 ONLY.

1. Enter LINK SELECT
2. On screen 10-010 enter a 'P' (for patch)
3. On screen 10-34P enter a '1'
4. On screen 10-26P enter 'IMPL' for the CSF name
5. On screen 10-27P enter: '1' for the drive number
   'VOL005' for the VOLID
6. On screen 10-28P enter the offset: '0B142'
7. On screen 10-29P enter the replacement data and CRC:

OFFSET	ORIGINAL DATA	REPLACEMENT DATA	CRC
0B142	7509C644	7409C644	775A

8. Use END for the next offset and then press FIELD EXIT and then ENTER.

After returning to READY INPUT you must perform steps 1-4 and 7-9 of the procedure, "How to Update Your System to the Current CSF Level, for 5247 Disk Systems," page 22.

(This page intentionally left blank.)

PROGRAM EDITING CONSIDERATIONS

When developing a program, editing problems can be avoided by periodically saving the program as source and reloading it from the source file.

One method is to maintain two copies during development; one copy is the normal internal format (Type 09), the other copy is the source format (Type 05). Occasionally load the source copy being sure it is the latest version. Free the internal copy and re-save it.

An alternative is to create a procedure that automatically saves as source, reloads, and re-saves. Be sure to use the latest version. Example:

```
LOAD pgm.name/volid
SAVE pgm.name.src/volid,SOURCE
LOAD pgm.name.src/volid
FREE pgm.name/volid
SAVE pgm.name/volid
FREE pgm.name.src/volid
```

'Pgm.name' is the name of the program and 'volid' is the volume that the program is located.

By using the above technique these situations can be avoided:

1. An Option 25 error 2000 occurs trying to change the program. This means that the program has been edited 31 times.

   The space used in memory and on the diskette grows each time the program is edited and replaced because deleted or changed lines are not removed.

2. An invalid error 0301 may occur executing a user defined function when the header line (DEF) of the function has been changed.

HOW TO CONVERT LOWER CASE LETTERS TO UPPER CASE

The following BASIC statements convert all the lower case letters in string S$ to upper case:

```
FOR I = 1 TO LEN(S$)
L=ORD(S$(I:I))
IF L>128 AND L<138 THEN S$(I:I)=CHR$(L+64)
IF L>144 AND L<154 THEN S$(I:I)=CHR$(L+64)
IF L>161 AND L<170 THEN S$(I:I)=CHR$(L+64)
NEXT I
```

If the string S$ is long, a faster method would be to just replace all letters individually with SREP$:

```
S$=SREP$(S$,1,"a","A")
S$=SREP$(S$,1,"b","B")
S$=SREP$(S$,1,"c","C")
     .
     .
     .
S$=SREP$(S$,1,"z","Z")
```

Or use a loop:

```
LET UPPER$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
LET LOWER$="abcdefghijklmnopqrstuvwxyz"
FOR I=1 TO 26
S$=SREP$(S$,1,LOWER(I:I),UPPER(I:I))
NEXT I
```

WHEN NOT TO USE PROTECT TO CLOSE FILES

The command, 'PROTECT filename,RELEASE ALL,' is used to close OPEN files when a power failure has occurred, when someone has turned off the computer while a program was running, or when the diskettes have been removed before the application ended. However, the PROTECT command should not be used if the program fails with an error or was interrupted by pressing CMD ATTN. In these situations type GO END or CLEAR PROC, and press ENTER, to end the program or procedure (note that GO END will continue with the procedure if one is active).

GO END or CLEAR PROC close the files "softly," updating the file headers with any new End of Data information. The PROTECT command does not update the file headers which causes any records that have been added to the open file(s) to be lost.

See the item titled "GO vs GO END vs CLEAR PROC" on page 63.

A TRAP ERROR LISTING A PROGRAM

1. If the program has a DATA statement that contains exactly 59 characters that are not surrounded by quotes, a TRAP error or unusual display may result when listing that statement.

   This can be avoided by enclosing character values listed in a DATA statement in quotes.

2. If a LIST command is issued to a nonexistent line number and the Scroll-up key is pressed, a TRAP error may occur.

   This can be avoided by not scrolling up unless the line listed is displayed. Instead, issue another LIST command to an existing line.

A TRAP ERROR RUNNING A PROGRAM

1. Using a dimensioned numeric variable as the counter of a FOR/NEXT loop can cause a TRAP error or Power-On Diagnostics (the self-diagnostic check done when the computer is first turned on). For example:

```
00010 DIM X(10)
00020 FOR X=1 TO 10
00030 NEXT X
00040 END
```

   This can be avoided by never using dimensioned variables as FOR/NEXT counters. It's not syntactically correct.

2. Using a MAT assignment in an IF/THEN statement with an ELSE clause may cause a TRAP error. For example:

```
00100 IF X=1 THEN MAT A=B ELSE MAT C=D
```

   This can be avoided by not using MAT assignments in IF/THEN statements with ELSE clauses.

NOTE: In general an invalid use of a MAT parameter may cause a TRAP error or power-on diagnostics.

3. Invoking a multiple line user-defined function that is coded at the end of the program may cause a TRAP error.

   This can be avoided by never ending a program with a user-defined function.

4. Writing or rewriting a numeric value as packed decimal (PD), that is exactly one digit larger than the FORM specification may cause a TRAP error or a value of zero to be written to the field. For example:

```
00100 LET X=1000
00110 WRITE #1,USING 120:X
00120 FORM PD 2
```

This may be avoided by checking values to be written to packed decimal fields to insure that they fit the FORM specification.

5. Using an ON (condition) IGNORE in an IF/THEN/ELSE statement can cause a TRAP error. For example:

```
00100 IF X=1 THEN ON ATTN IGNORE ELSE GOTO 5100
```

This can be avoided by putting the ON (condition) IGNORE in a separate statement.

6. Using a formatted READ statement with no variables may cause a TRAP error. For example:

```
00100 READ #1,USING 110:
00110 FORM . . .
```

This may be avoided by omitting the USING parameter from the READ statement or including at least one variable.

7. Using the SREP$ System Function may cause a TRAP error if the replacement string is longer than the substring being replaced. For example:

```
00100 LET X$="ABCDEF"
00110 LET X$=SREP$(X$,1,"CD","XYZ")
```

This may be avoided by ensuring that the replacement string is less than or equal to the length of the substring being replaced.

To circumvent this problem, use the following code:

```
00080 LET R$="CD"
00090 LET Z$="XYZ"
00100 LET P1=0
00110 LET P2=POS(X$,R$,P1)
00120 IF P2=0 THEN 160
00130 LET X$=X$(1:P2-1)&Z$&X$(P2+LEN(R$):LEN(X$))
00140 LET P1=P2+LEN(R$)
00150 GOTO 110
00160 LET Y$=X$ ! CONTINUE PROCESSING
```

TRIGONOMETRIC FUNCTIONS SIN AND COS INCORRECT

The trigonometric functions, SIN and COS give incorrect results in some cases. When the result should be -1 the computer gives a result of +1. For example, COS(PI) returns +1 when it should return -1.

This may be avoided by using the following substitutions:

```
SIN(X)  use  (2*(COS(PI/4-X/2)**2)-1)

COS(X)  use  (2*(COS(X/2)**2)-1)
```

THE SYSTEM FUNCTION SRCH WORKS IMPROPERLY

1. If the array being searched has more than 255 elements in it, the SRCH function fails to find elements at locations that are multiples of 256.

   This may be avoided by not searching arrays with more than 255 elements, or check the 256th, 512th, . . . elements separately from the SRCH.

2. If the SRCH function is used as part of a calculation, the results are unpredictable. For example:

   ```
   00100 LET X=SRCH(A,Y,1)+2
   ```

   This may be avoided by doing the SRCH in a separate line:

   ```
   00090 LET X1=SRCH(A,Y,1)
   00100 LET X=X1+2
   ```

3. If the argument of the SRCH function is a subscripted variable, SRCH may have unpredictable results. For example:

   ```
   00100 IF SRCH(A,B(1),1) . . .
   ```

   This may be avoided by assigning the subscripted argument to a non-subscripted variable:

   ```
   00090 LET Y=B(1)
   00100 IF SRCH(A,Y,1) . . .
   ```

4. Error 0405 occurs using SRCH that specifies a starting row number larger than 255.

   This may be avoided by not searching arrays of more than 255 elements.

CONVERSION TO SYSTEM/36 TECHNIQUES

The following guidelines have been developed to assist the Datamaster programmer in converting programs and files to the System/36. To convert the Datamaster to the System/36 involves three areas of consideration: converting programs, converting files, and programming differences. These guidelines were developed mainly to provide a tool to assist the programmer with the conversion process. For additional information regarding the BASIC programs and formats on the System/36, refer to the publication "PROGRAMMING WITH BASIC," SC21-9003.

CONVERTING DATAMASTER BASIC PROGRAMS TO SYSTEM/36

Steps to be Completed on the Datamaster

1. Prepare an H-Exchange (diskette type 2D) or Basic-Exchange (diskette type 1D) diskette using the CSF PREPARE

2. Load the program to be converted

   LOAD program/volid

3. Save this program to a new file as source

   SAVE program.src/volid,SOURCE

4. Use a BASIC program to copy the source version to the exchange diskette.

Example of a BASIC program to copy the source to the exchange diskette:

```
00010 OPTION BASE 1
00020 DIM A$*255,B$(4)*80
00030 OPEN #1:"name=progid.src/volid/1",DISPLAY,INPUT
00040 OPEN #2:"name=filename//2,size=200000,recl=80,format=hx",
      INTERNAL,OUTPUT
00050 GOSUB FIRST
00060 !
00070 !
00080 RDLOOP: LINPUT #1: A$ EOF WHOA
00090 !
00100 !
00110 IF LEN(A$)>80 THEN LET B$(1)=A$ ELSE 130
00120 GOTO PUT
00130 LET B$(1)=A$(1:79)&"&"
00140 IF LEN(A$)<157 THEN LET B$(2)="&"&A$(80:157)&"&"
      ELSE 170
00150 IF LEN(A$)<235 THEN LET B$(3)="&"&A$(158:235)&"&"
      ELSE 190
```

```
00160 IF LEN (A$)<236 THEN LET B$(4)="&"&A$(236:LEN(A$))
      ELSE PUT
00170 LET B$(2)="&"&A$(80:LEN(A$))
00180 GOTO PUT

00190 LET B$(3)="&"&A$(158:LEN(A$))
00200 GOTO PUT
00210 !
00220 !
00230 PUT: FOR X=1 TO 4
00240 IF B$(X)="" THEN 280
00250 WRITE #2:,USING 260:B$(X)
00260 FORM C 80
00270 NEXT X
00280 MAT B$("")
00290 GOTO RDLOOP
00300 WHOA: GOSUB LAST
00310 CLOSE #1:
00320 CLOSE #2:
00330 STOP
00340 FIRST: WRITE #2, USING 350:"// COPY
      NAME-membername,LIBRARY-S"
00350 FORM C 80
00360 RETURN
00370 LAST: WRITE #2: USING 380:"// CEND"
00380 FORM C 80
00390 RETURN
```

NOTE: This sample program inserts a //COPY statement as the first record and a // CEND statement as the last record of the exchange file that the 'TOLIBR' command on the System/36 will use. These statements are unnumbered and must begin in position 1. It also breaks a 255-character Datamaster BASIC statement into multiple 80-character System/36 BASIC statements with the '&' continuation character in position 80.

At this point, the file on the exchange diskette is ready to be converted to the System/36.

Steps to be Completed on the System/36

1. Copy the diskette file to the System/36 using the 'TOLIBR' command:

   TOLIBR filename,I1,,,library name

   filename=name of file created on Datamaster
   library name= Any user library on the System/36

   You now have a BASIC source member (program) in the user library specified in the 'TOLIBR' command.

2. Sign on to System/36 BASIC mode with the following command:

   BASIC

3. Load the BASIC source member (program) from the System/36 user library:

   LOAD membername,library name,SOURCE

   membername = program name
   library name = System/36 user library

As the source program attempts to load on the System/36, you may encounter syntax errors. Several programming considerations which may help prevent some of these syntax errors include:

- The SHIFT statement has been replaced by the 'O' attribute in Full Screen Processing to allow entry of lowercase characters into an input field (See INPUT FIELDS statement in "PROGRAMMING WITH BASIC," Chapter 17). If entering programs in BASIC, use the Command 3 to allow entry of lowercase characters.

- File names may only be 8 characters long and must not contain VOLID or DRIVE number.

- Program names may not contain periods (.).

- The following reserved words have been added on the System/36:

ALTERNATE	GET	RAD
ATTRIBUTE$	IMAGE	RELEASE
BASE	INDIC	RETCODE$
BEGIN	INF	SEARCH
BREAK	INVP	SPREC
CNVRT$	IP	STANDARD
COLLATE	LPREC	STARTLINE
CURROW	LWRC$	STOP$
CURCOL	MOD	STREAM
DEBUG	MSG$	SUM
DEG	NATIVE	TIMER
DUPKEY	NOT	UPRC$
FILENUM	OFF	UPSI$
FORMAT	PROCLVL	USERID$
FP	PRTZO	WAITIO
FREE	PUT	

- Remarks may be indicated by ¦ (vertical bar) as well as by REM or ! (exclamation point). Remarks may be used on any statement except DATA, FORM, or IMAGE statements.

- The IMAGE statement has been added to allow for more control of the printed output than was available in the FORM statement. IMAGE allows for left alignment, right alignment, or centering of character data.

- To allow a printer to start printing a report without closing the printer file, a // PRINTER OCL statement with DEFER-NO should be used. Otherwise, the spool file will not start printing until the printer file is closed or the program ends.

- The SIZE parameter of the OPEN statement can be specified in either blocks or records rather than a number of characters.

- A READ statement may be required to detect EOF if the EOF clause was used on an OPEN, RESTORE, or DELETE statement on the Datamaster.

- Printer values of LPI and CPI cannot be changed with the PRINT statement within a program.

  The default for the OPTION statement on the System/36 is short precision which gives only six significant digits. The Datamaster uses long precision only.

  Utilities to sort, copy, and build indexed files are provided by the System/36 SSP. The SORT CSF must be replaced by OCL and System/36 Specifications. The COPY CSF should be replaced with the COPYDATA procedure to copy one file to another. The INDEX CSF to rebuild an index need not be run if the file has only one key field. If multiple indexes exist for the file, BLDINDEX must be run for the alternative indexes for the file (the data file must already have an index for the BLDINDEX to be run). On the System/36 the index is an integral part of the file.

  Syntax errors can be handled in any one of the following ways:

  A. Correct the error and press ENTER

  B. Make the statement a comment by inserting either a REM, ! (exclamation point), or ¦ (vertical bar) and space in front of the statement

C. Press the roll up key to delete the statement

4. Save the program as a subroutine library member (subroutine library members are where the System/36 BASIC programs normally reside):

   SAVE membername,library name

   membername = program name
   library name = System/36 user library name

   NOTE: Since source and subroutine members on the System/36 are two separate members, the same member name may be used for both.

5. Load and run the program as a System/36 subroutine member:

   LOAD membername, library name
   RUN

   membername = program name
   library name = System/36 user library name

TRANSFERRING DATAMASTER DATA FILES TO SYSTEM/36

The following steps outline the transfer of Datamaster data files to the System/36:

1. Prepare an H-Exchange (diskette type 2D) or Basic-Exchange (diskette type 1D) using PREPARE CSF on the Datamaster

2. If the record length is:

   - less than or equal to 128 bytes (Basic-Exchange)
   or
   - less than or equal to 256 bytes (H-Exchange)
   and
   the file is contained on a single diskette, use the CSF COPY to copy the Datamaster internal file (Type 04) to the exchange diskette

3. If the record length is:

   - greater than 128 bytes (Basic-Exchange)
   or
   - greater than 256 bytes (H-Exchange)
   or

the file cannot be contained on a single diskette, the file will need to be reformatted by a BASIC program in order to transfer the file to the System/36.

If the file needs to be reformatted, be sure to specify FORMAT=HX or FORMAT=BX in the OPEN statement and to use a simple file name (no periods).

NOTE: Numeric field types 'L' and 'N' should be converted using a BASIC program prior to transfer if the files are to be accessed by RPG II or DFU (Data File Utility).

The number of records in the file can be used to determine if multiple diskettes will be required.

Diskette Type	Record Length	Approximate # of Records
1D	128	1896
2D	256	1896
2D	128	3792

Only Datamaster internal format (Type 04) files can be transferred. (Type 07 (Index file), Type 08 (Index - keys in overflow), or Type 20 (Word Processing Document file) format files cannot be transferred.)

4. The following steps may be used to load the transferred files from the exchange diskette onto the System/36:

   Use the System/36 'TRANSFER' procedure to load the file from the exchange diskette to the System/36. If the file was reformatted because the number of records in the file could not be contained on a single diskette, use the 'TRANSFER' procedure with the Add to Existing File parameter for the second and all remaining diskettes for the given file.

   If the file was reformatted because the record length was too long, (a) use the 'TRANSFER' procedure to load onto the System/36 then, (b) write a BASIC program to reconstruct the file to the original record length from the individual record written to the diskette.

   If the file was INDEXED or DIRECT on the Datamaster, use the System/36 'COPYDATA' procedure to create the correct file type and (if indexed) supply appropriate key information.

   If multiple indexes existed on the Datamaster use 'COPYDATA' for the primary index file, then use 'BLDINDEX' to create the alternative indexes.

SYSTEM/36 DIFFERENCES

Programming Items

- On READ statements for an indexed file opened for random update the KEY= parameter must be entered if records are being added.

- 'ON' condition statements performed inside a function remain in effect outside the function.

- GOTO and GOSUB may not be used to enter or leave a function.

- The blinking cursor may not be available on certain displays; therefore, you may want to change any underline fields to column separator field attributes.

- The cursor advance pattern is from left to right, then top to bottom of the screen. If you want the cursor to move from top to bottom, then left to right, you must arrange your INPUT/PRINT FIELDS statements to handle this.

- Datamaster characters are not supported on the System/36.

- The TAB function of the PRINT statement allows row and column positioning.

Procedures

- A procedure is a library member (type P) on the System/36

- DROP is replaced on the System/36 with either the CLOSE #X: FREE and a 'BLDFILE' procedure or a 'DELETE' procedure and a 'BLDFILE.'

- BASIC programs may be loaded and run from an SSP procedure with the BASICR procedure.

Files

- Duplicate file names may exist if they have different creation dates even if the NEW is specified in the open statement.

- Record lengths are the physical record length, not the record length plus 1.

- Fields that were created by a BASIC program with FORM types of 'L,' 'S,' or 'N' can only be read by BASIC programs. Zoned decimal (ZD) form should be used for numeric fields if the files are to be used by other programs.

HOW TO TRANSFER DATA AND PROGRAMS DATAMASTER ---- SYSTEM/34

The following steps explain how to transfer data files and BASIC programs from the Datamaster to the System/34 and from the System/34 to the Datamaster:

Data Files - Datamaster to System/34:

1. Prepare an H-Exchange (diskette Type 2D) or Basic-Exchange (diskette Type 1) diskette using the PREPARE CSF on the Datamaster

2. Create the Exchange file by using a BASIC program or the COPY CSF to copy the Datamaster internal format (Type 04) Datamaster file to the Exchange diskette. A BASIC program has to be used if more than one Exchange diskette is required for a single file

   If a BASIC program is used be sure to specify FORMAT=HX or FORMAT=BX in the OPEN statement and use a simple file name (no periods).

   NOTE: the record length must be less than or equal to 128 for Basic-Exchange or 256 for H-Exchange.

3. Use TRANSFER or $COPY on the System/34 to copy the data from the Exchange diskette to a System/34 file.

Data Files - System/34 to Datamaster:

1. Prepare an H-Exchange (diskette Type 2D) or Basic-Exchange (diskette Type 1) diskette. This can be done on the Datamaster or the System/34. If done on the System/34, be sure to specify a VOLID

2. Use TRANSFER or $COPY on the System/34 to copy the data from the System/34 file to the Exchange diskette. If more than one diskette is used for a single file, a program has to be written on the Datamaster to integrate them back into one file. The Datamaster does not support multiple volume files

3. Use a BASIC program or the COPY CSF on the Datamaster to copy the Exchange file to a Datamaster internal format file (Type 04).

PROGRAMS - DATAMASTER TO SYSTEM/34:

1. Prepare an H-Exchange (diskette Type 2D) or Basic-Exchange (diskette Type 1) diskette using the PREPARE CSF on the Datamaster

2. Load the program to be transferred and save it to a new file as source:

   LOAD pgm/volid

   SAVE pgm.src/volid,SOURCE

3. Use a BASIC program to copy the source version of the program to the Exchange diskette

   This program must break up program lines longer than 80 characters into multiple records. Each part of a line must have an '&' concatenated on the end, except the last part, and on the beginning, except the first part. For example, if A$ contains the current program line read from the source file, the following statements break it up into 79-character-long segments if required by its length and write each segment into the specified file:

```
00050 OPTION BASE 1
00060 DIM A$*255
00070 OPEN #2:"NAME=input.file/volid/1",DISPLAY,INPUT
00080 OPEN #1:"NAME=output/hx/1,SIZE=100000,RECL=80,
      FORMAT=HX",INTERNAL,OUTPUT,SEQUENTIAL
00090 LINPUT #2:A$ EOF 170
00100 IF LEN(A$)<=80 THEN GOTO 150
00110 WRITE #1,USING 120:A$(1:79)&"&"
00120 FORM C 80
00130 LET A$="&"&A$(80:LEN(A$))
00140 GOTO 100
00150 WRITE #1,USING 120:A$
00160 GOTO 90
00170 CLOSE #1:
00180 CLOSE #2:
```

4. The BASIC program that creates the Exchange file can also put the following two OCL commands before and after the BASIC statements for the TOLIBR command on the System/34 to use

```
// COPY FROM-DISK,TO-libraryname,LIBRARY-S,
     NAME-membername,RETAIN-P,FILE-filename

(BASIC statements)

// END
```

5. If TRANSFER is used the above OCL is not included. Example:

   ```
   TRANSFER filename
   ```

6. If TOLIBR is used with the above OCL included in the Exchange file, use the following command on the System/34:

   ```
   TOLIBR filename,I1,,,libraryname
   ```

7. Load the program in BASIC mode on the System/34 from the source library and save it as a subroutine library member (where System/34 BASIC programs normally reside). Use the following commands:

   ```
   LOAD membername,libraryname,SOURCE
   SAVE membername,libraryname
   ```

PROGRAMS - SYSTEM/34 TO DATAMASTER

1. Prepare an H-Exchange (diskette Type 2D) or Basic-Exchange (diskette Type 1) diskette. This can be done on the Datamaster or the System/34. If done on the System/34, be sure to specify a VOLID

2. Load the BASIC program from the library and save it as source to the source library. Use the following commands:

   ```
   LOAD membername,libraryname
   SAVE membername,libraryname,SOURCE,80
   ```

3. Use the $MAINT utility to convert the source member to the Exchange diskette (record length of 80)

   ```
   // LOAD $MAINT

   // FILE NAME-filename,UNIT-I1,LABEL-filename,
        PACK-diskettevolid

   // RUN

   // COPY FROM-libraryname,TO-DISK,FILE-filename,RECL=80,
        NAME=sourcemembername,LIBRARY-S,BASIC-YES

   // END
   ```

4. Use a BASIC program on the Datamaster to convert the Exchange file to a Datamaster source file (Type 05). Note that any program lines that are greater than 80 characters are in multiple records on the Exchange diskette. The following program statements are an example of how to create a single record that is

PRINTed to the source file. The record read is in A$;
the record to be printed is in B$

```
00020 DIM A$*80,B*255
00030 OPEN #1:"NAME=src/volid,SIZE=20000,RECL=255",
     DISPLAY,OUTPUT
      .
      .
00090 IF A$(1:1)="&" THEN A$=A$(2:LEN(A$)) ! STRIP '&'
00100 IF A$(1:2)="//" THEN READNXT ! SKIP OCL RECS
00110 LET B$=B$&A$
00120 IF LEN(A$)<80 THEN PRTIT
00130 IF A$(80:80)><"&" THEN PRTIT
00140 LET B$=B$(1:LEN(B$)-1) ! STRIP '&' FROM END
00150 GOTO READNXT
00160 PRTIT: PRINT #1:B$
00170 LET B$=""
00180 GOTO READNXT
```

5. Once the program has been put into a source file, it needs to be loaded and saved as a program internal format file (Type 09). Use the following command to load it:

   LOAD src/volid

   The program is listed on the display, line-by-line, as it is loaded. If a syntax error occurs, correct the error or make the line into a remark for later analysis.

   Once the program is loaded use the following command to save it to a new type 09 file:

   SAVE prog.id/volid

HOW TO USE AN ADDRESS OUT FILE

One of the options when sorting a file is address out. This creates a smaller output file containing the relative record numbers (in format PD 3) of the master file in the order it was sorted. This sort has several advantages over a full record sort: It runs much faster (especially on large files), the output file is much smaller (4 bytes/master record), and it does not change the order of the master file, thus, leaving any index files valid.

Using an address out file needs a few extra statements in the program to access the master file in the sorted order. The following statements are an example of how to use an address out sort file:

```
00030 OPEN #1:"NAME=addr.out/vol",INTERNAL,INPUT
00040 OPEN #2:"NAME=master/vol",INTERNAL,INPUT,RELATIVE
      .
      .
      .
00100 READ #1,USING 110:X EOF DONE
00110 FORM POS 1,PD 3
00120 READ #2,USING 130,REC=X: . . .
00130 FORM . . .
```

Line 30 opens the address out file that is read sequentially.

Line 40 opens the master file that is read relatively using the values read from the address out file.

Line 100 reads the address out records into the numeric variable X. The end-of-file exit is on this read because it is the file that is being read sequentially. When end-of-file is reached, all the master records have been processed.

Line 110 specifies the format of the address out file.

Line 120 reads the record at position X in the master file.

WRONG RECORD READ OR DELETED

When accessing a file using an index file, the computer does not verify that the index file entry matches the key field in the master record. Therefore, the wrong record could be read, changed, or deleted.

The key field of the master record can be changed using sequential access or by accessing the record using another index file. The index file can be made invalid by copying the master file removing deleted records and not creating a new index file.

To avoid this problem, be sure to recreate the index file any time the master file is changed using sequential access or using another index file, and any time it is copied removing deleted records (see item "How Often Should INDEX be Run," page 11).

In addition any time you read a file by key, verify that the key field in the master record read matches the key used to access it.

A TRAP ERROR DURING FILE OPERATIONS

1. Writing or rewriting a numeric value to a packed decimal (PD) field that is exactly one digit too large can cause a TRAP error or a value of zero to be written to the field. For example:

   ```
   00100 LET X=1000
   00110 WRITE #1,USING 120:X
   00120 FORM PD 2
   ```

   This may be avoided by checking values to be written to packed decimal fields to insure that they fit.

2. Using a formatted READ statement with no variables may cause a TRAP error. For example:

   ```
   00100 READ #1,USING 110:
   00110 FORM . . .
   ```

   This may be avoided by omitting the USING parameter from the READ statement or including at least one variable.

READ SEARCH= FAILS TO FIND RECORD

Doing a READ with the SEARCH= parameter may fail to find a record that was added to the file since the last time INDEX was run.

This may be avoided by building a new index file using INDEX (see item "How Often Should INDEX be Run," page 27), or by using the KEY= or KEY>= parameters to find the record.

If the KEY= or KEY>= parameters are used, the argument variable must be the same length as the record's key.

If the KEY = parameter is used, the key field of the record should be included in the variable list so that the key field of the record read can be checked against the KEY>= argument to insure that the correct record has been read (see items "How to Access Records with Duplicate Keys," page 58, and "How to Access Records with a Partial Key," page 58).

READ REC=0 "WORKS" ON 5247 DISK SYSTEMS

If the computer has a 5247 Disk Unit Attachment Feature (the logic card that permits attaching the disk), executing a READ to a diskette or disk file with a REC= parameter that is zero does not cause a NOREC error or exit. It reads a record, but the data read is unpredictable.

To avoid this problem the program should check the REC= argument to insure that it is not zero. If it is zero, take whatever action is appropriate for the application.

HOW TO USE A VARIABLE FILE NAME

It is useful to be able to open a file whose name is determined by a control file record, operator input, or work station ID. This requires making the name of the file in the OPEN statement a variable. A portion of the OPEN statement is a character string enclosed in quotation marks. This can be entered as a literal string or partially replaced by a character string. For example, if the name of the file to be opened is the value of a variable NAME$, the OPEN statement could be coded:

```
OPEN #1:"NAME="&RTRM$(NAME$)&"/vol",INTERNAL,INPUT
```

Other portions of the open string can be made variable in a similar way. For example, on a new file the initial size can be set using the value of the numeric variable SZ:

```
OPEN #1:"NAME=file/vol,SIZE="&STR$(SZ)&",RECL=63",
   INTERNAL,OUTPUT
```

READ THE DISKETTE OR DISK DIRECTORY

Many programmers would like to be able to read the directory information about a file in a program. This is not possible except for the information available through the System Functions FILE(x), FILE$(x), FREESP(x), KLN(x), KPS(x), REC(x) and RLN(x) where x is the file reference number used in the OPEN statement (1-127). These functions provide the following information:

FILE(x) returns the status of the file.

```
-1 - File is not open
 0 - Normal end of I/O operation
10 - EOF during Input
11 - EOF during Output
20 - Transmission Error during Input
21 - Transmission Error during Output
```

FILE$(x) returns a character string containing the file name, VOLID and device number; the values are separated by a slash. It returns a null string if the file is not opened.

FREESP(x) returns the number of 512 byte blocks available on the diskette or disk on which the file is allocated. It returns a -1 if the file is not opened or the device type is not diskette or disk (for example, the printer).

KLN(x) returns the key length of the file. It returns a -1 if the file is not opened or if it is not opened as an indexed file (KFNAME= specified in the OPEN statement).

KPS(x) returns the key field starting position in the master record. It returns a -1 if the file is not opened or if it is not opened as an indexed file (KFNAME= specified in the OPEN statement).

REC(x) returns the record number of the last record processed in the file. It returns 0 if no records have been processed. It returns a -1 if the file is not opened, if it is an indexed file, or if it is a display file.

RLN(x) returns the record length of the file. It returns a -1 if the file is not opened.

W TO UPDATE A FILE USING TWO KEYS

the master file has two index files, which simultaneously
cess a master record, how can the file be updated using
ther key? If the computer has the 5247 Disk Unit, this
n be done by opening the file twice with the share-for-up-
te (SHRU) parameter, but on a diskette only computer, the
le cannot have two active opens for update. The following
atements illustrate a technique that can be used:

```
00030 OPEN #1:"NAME=mstr/vol,KFNAME=ix01/vol,SHR",INTERNAL,
      INPUT,KEYED
00040 OPEN #2:"NAME=mstr/vol,KFNAME=ix02/vol,SHR",INTERNAL,
      OUTIN,KEYED
      .
      .
      .
00100 IF PATH=2 THEN GOTO PATH2 ! GOTO PATH2 IF IX02 USED
00110 READ #1,USING 120,KEY=KEY1$:KEY2$, . . .
00120 FORM POS 6,C 5, . . .
      .
      .
      .
00200 GOTO REWRT
00210 PATH2: READ #2,USING 220,KEY=KEY2$: . . .
00220 FORM . . .
      .
      .
      .
00300 REWRT: REWRITE #2,USING 310,KEY=KEY2$: . . .
00310 FORM . . .
```

Line 30 opens the master file using the first index for input only, shared.

Line 40 opens the master file using the second index for input or update, shared. This is the open that is used to update the file accessed using either key.

Line 100 checks to see which index file to use to access the file. The value of PATH is determined by an appropriate means for the application.

Line 110 reads the file using the first index. It also reads in that record's second key (its length is five in this example) that will be used to update the file at line 300.

Line 200 goes to update the record after the appropriate processing has been done.

Line 210 reads the file using the second index.

Line 300 updates the record using the second key regardless of which key was used to read it.

NOTE: If any records are added to the master file, only the index file IX02 is updated. The INDEX CSF has to be rerun to generate a correct and complete IX01 index file.

UNFORMATTED FILE I/O

Using unformatted READ and WRITE statements (no USING parameter) to a disk or diskette file is not recommended. The record created by an unformatted read has 3 bytes of computer data in the first 3 positions. This can cause unexpected results when attempting to read those records using a formatted READ.

This may be avoided by always specifying the USING parameter and using a FORM statement.

OPEN SHARING MATRIX

The following matrix shows which types of opens are permitted by a second or subsequent work station depending on the sharing status and type of OPEN coded. This matrix includes the disk and diskette sharing options. If the computer does not have the 5247 Disk Unit, the SHRU option is not available.

There are three types of OPEN: INPUT, OUTPUT, OUTIN. For this matrix OUTPUT and OUTIN have the same results.

There are four types of sharing status that can be specified:

NOSHR - No file sharing (this is the default)
SHRI - Share for Input only
SHR - Share
SHRU - Share for update (5247 Disk systems only)
File MUST be opened SHRU to use the RELEASE command

1st \ 2nd		NOSHR		SHRI		SHR		SHRU	
		I	O/U	I	O/U	I	O/U	I	O/U
NOSHR	I	N	N	N	N	N	N	N	N
	O/U	N	N	N	N	N	N	N	N
SHRI	I	N	N	Y	N	Y	N	Y	N
	O/U	N	N	N	N	Y	N	Y	N
SHR	I	N	N	Y	Y	Y	Y	Y	Y
	O/U	N	N	N	N	Y	N	Y	N
SHRU	I	N	N	Y	Y	Y	Y	Y	Y
	O/U	N	N	N	N	Y	N	Y	Y

I = Opened for INPUT
O/U = Opened for OUTPUT or OUTIN (update)
N = 2nd OPEN causes an error
Y = 2nd OPEN succeeds
1st = Most restrictive of current OPEN(s)
2nd = New OPEN being executed

HOW TO ACCESS RECORDS WITH DUPLICATE KEYS

The Datamaster supports index files with duplicate keys. However, when using KEY= only the first record that satisfies the KEY parameter is accessed. Issuing the same READ again gets the same master record again. The code below shows how to access the second and subsequent records with the same key:

```
00010 OPEN #1:"NAME=mst/vol,KFNAME=indx/vol",INTERNAL,
      INPUT,KEYED
      .
      .
      .
00100 READ #1,USING 110,KEY=K$: . . .
00110 FORM . . .
00120 GOTO 160
00130 NXT: READ #1,USING 140:KY$, . . . EOF NXTKEY
00140 FORM C 5, . . .
00150 IF KY$><K$ THEN GOTO NXTKEY
00160 REM PROCESS THE RECORD
      .
      .
      .
00200 GOTO NXT
```

Line 100 reads the first occurrence of the record with the key of K$.

Line 120 skips the read of second and subsequent records to process the first record read.

Line 130 reads the second and subsequent master records that may have the same key. Note that the KEY= parameter has been omitted and that the key field is being read as part of the data (in this example the key length is five).

Line 150 checks to insure that the key field of the second or subsequent record is equal to the key value used to access the first record. If it is not, it branches to the code that requests or determines the next key value to use.

Line 200 returns to read another record.

HOW TO ACCESS RECORDS WITH A PARTIAL KEY

The Datamaster supports accessing index files with a partial key; however, when using KEY>= only the first record that satisfies the KEY parameter is accessed. Issuing the same READ again gets the same master record again. The code below shows how to access the second and subsequent records

with the same partial key:

```
00010 OPEN #1:"NAME=mst/vol,KFNAME=indx/vol",INTERNAL,
      INPUT,KEYED
      .
      .
      .
00090 LET K$=RPAD$(K$,5)
00100 READ #1,USING 110,KEY>=K$:KY$, . . .
00110 FORM C 5, . . .
00115 IF KY$(1:3)><K$(1:3) THEN GOTO NOTFND
00120 GOTO 160
00130 NXT: READ #1,USING 140:KY$, . . . EOF NXTKEY
00140 FORM C 5, . . .
00150 IF KY$(1:3)><K$(1:3) THEN GOTO NXTKEY
00160 REM PROCESS THE RECORD
      .
      .
      .
00200 GOTO NXT
```

Line 90 insures that the length of the key parameter is equal to the key length by adding trailing blanks (in this example the key length is five).

Line 100 reads the first occurrence of a record with a key greater than or equal to the value of K$. Note that the key field is being read as part of the data.

Line 115 checks to insure that the first record found has a key field that matches the first three characters, in this example, of the parameter value used in line 100. If it is not, it branches to handle the "record not found" condition.

Line 120 skips the READ of second and subsequent records to process the first record read.

Line 130 reads the second and subsequent master records that may have the same key. Note that the KEY>= parameter has been omitted from this read.

Line 150 checks to insure that the key field of the second or subsequent record has a key field that matches the first three characters of the parameter value used to access the first record. If it is not, it branches to the code that requests or determines the next key value to use.

Line 200 returns to read another record.

DATA FILE PERFORMANCE CONSIDERATIONS

Sequential Access

If memory is available when the file is opened, the computer allocates a buffer equal to a full diskette track (about 7.5K) which minimizes the number of accesses to the file. Therefore, high usage files should be opened first to take advantage of this capability. This space can be taken back by the computer (to the minimum buffer of 512 bytes) if the program requires the memory in a subsequent operation.

When specifying the initial size of a file (using the SIZE= parameter) attempt to make it large enough for the data. Although the computer may allocate up to 100 additional extents, these extents can be scattered on the diskette or disk which significantly degrades performance.

NOTE: Extents are allocated as 10% of the original allocation or 512 bytes which ever is larger.

Indexed Access

The most significant performance consideration for indexed files is the number of records added to the file since the last time the INDEX CSF was run. The key records are written to an overflow area of the index file and are not in sorted order. Therefore, accessing those records is slower than accessing records in the original part of the key file.

Performance can be improved by regularly running the INDEX CSF to create a new index file. This operation can be in a procedure so that the user does not have to take any special steps (see item "How Often Should INDEX be Run," page 11).

For large indexed files, specifying a key work area (the KW= parameter) can improve performance. It allocates an area in memory to store selected index records from the sorted portion of the index file. This allows the computer to do its initial index search in memory, eliminating some accessing of the index file. If you specify a key work area of 65,535 bytes (KW=65535), the computer allocates only the space necessary based on the size of the index file. The improvement in performance is greater for large files where the index file spans several tracks.

As with sequential files, when specifying the initial size of a file (using the SIZE= parameter), attempt to make it large enough for the data. Although the computer allocates up to 100 additional extents if needed, these extents can be scattered and significantly degrade performance. This is also true of the index file; however, the initial size of the index file is determined by the number of used records (active and deleted) in the master file when the INDEX CSF

is run. Therefore, it is important to run INDEX any time records have been added to the file.

There should be an improvement in performance on a diskette system if the master and index file are on separate diskettes. However, the improvement is small compared with the above items.

PERFORMANCE OF THE DISK

How much improvement in performance does the 5247 Disk Unit have? Although the disk unit has faster access and higher data transfer rate than diskette, the improvement in performance of an application using the disk depends on how much and what type of file accessing is done.

As discussed in the item titled "Data File Performance Considerations" on page 55 if sequential access is used, 7.5K is transferred to the buffer at a time. All accesses of these records then take place in memory, which is the same speed whether the data is read from disk or diskette.

Processing and printing take the same amount of time independent of the disk or diskette.

Applications with a lot of index file access should show the most improvement in performance.

The major improvement is not the running of a single application, but the capability to run four applications on four work stations all sharing the disk unit without a decrease in performance. In fact, if the four applications are not contending for the same files, they should run faster than they would on a diskette computer.

There is one situation where the computer runs slower. If an application has a file opened SHRU (share for update), it runs slower than if the file is opened using any of the other sharing options. There is additional overhead required by SHRU: the sectors used by the application have to be "locked" and "unlocked," and the file header has to be updated with each access since another work station may be using it. Therefore, only use SHRU when absolutely required by the application.

HOW TO DETERMINE THE RECORD NUMBER OF THE LAST RECORD

When using relative access it is necessary to know the record number of the last record in the file in order to add new records (since it must be specified in the WRITE statement). The following statements may be used to

determine the number of the next record in the file
(subtract one to get the number of records in the file):

```
00030 OPEN #1:"NAME=file/volid",INTERNAL,OUTPUT
00040 WRITE #1:" "
00050 R=REC(1)
00060 CLOSE #1:
00070 OPEN #1:"NAME=file/volid",INTERNAL,OUTIN,RELATIVE
00080 DELETE #1,REC=R:
```

Line 30 opens the file for output, sequential.

Line 40 adds a temporary record to the file.

Line 50 saves the record number of the record added (the value of the System Variable REC) in the variable R. R-1 would be the number of records in the file (including deleted records).

Line 60 closes the file.

Line 70 opens the file for update, relative.

line 80 deletes the added record; it can be added again by a subsequent WRITE statement.

GO VS GO END VS CLEAR PROC

If a program being executed from a procedure is interrupted by an error, pressing CMD ATTN, or a PAUSE statement, three options are available:

1. GO or GO xxxxx causes the interrupted program to resume execution at the current (if error) or next line, or at the line number specified by xxxxx.

2. GO END terminates the current program and closes any files opened in that program, but does not end the procedure. The next procedure command is now executed as if the program ended normally.

3. CLEAR PROC terminates the current program and procedure, closes all files opened in the program, closes all active procedure and sub-procedure files, and the computer displays READY INPUT.

If PROC INPUT is displayed when the interrupt occurs:

1. GO causes the next procedure command to be executed.

2. GO END closes only the active PROC or SUBPROC file. It returns control to the procedure command following the most recent SUBPROC command, or if none, the computer displays READY INPUT.

3. CLEAR PROC closes all procedure and sub-procedure files and the computer displays READY INPUT.

FALSE ERROR MESSAGE FROM PROCEDURE

When running procedures that use PROCERR RETURN to suppress errors or reset the System Variable ERR to 0, it is possible to get an error message displayed that is not the true cause of the problem.

Some errors are not trapped by PROCERR RETURN (e.g. 1008, 2104, 4000 and 4001). Therefore, if an error occurs that is trapped by PROCERR RETURN and is not handled properly by the procedure (by analysing ERR), the procedure continues until an error, which cannot be suppressed, occurs. This masks the true cause of the problem.

To avoid this be sure to reset the return with a PROCERR STOP command as soon as the steps for which errors are being trapped are completed. Don't use a 'blanket' PROCERR RETURN in procedures.

Also note that any PROCERR command issued in a procedure remains in effect for all subsequent procedures and sub-procedures, whether chained to from a program or invoked from a procedure. Therefore, it is a good idea to begin any procedure with PROCERR STOP.

In order to reset the System Variable ERR to zero without suppressing errors include a PROCERR RETURN followed by a PROCERR STOP in the procedure.

NO ERROR MESSAGE WHEN RUNNING A DROPPED PROC

If a Proc file is DROPped, you will not get an error message when the Proc is called either by a PROC or SUBPROC command. This is because a Proc file is a data file that is executed sequentially and will always return to READY INPUT when it reaches the end of file. The DROPped Proc file has a file label and an end of file marker; therefore, when the PROC or SUBPROC command is issued, the system returns to READY INPUT and does not display an error as one has not occurred.

A DEL COMMAND WITH A COMMENT DELETES ALL SUBSEQUENT LINES

If a DEL command is issued from a procedure or the keyboard and it has a comment on it (an exclamation point, '!'), the command will delete the line specified plus all following lines of the program in memory.

To avoid this do not use comments on DEL commands.

REMOVING DISKETTES WHILE A PROCEDURE IS ACTIVE

It is sometimes necessary to remove a diskette during the execution of a procedure (especially during copy operations). However, if the diskette to be removed contains the active procedure, an error 4000 or 4001 may occur. To avoid this have the current procedure issue a PROC to a procedure on the diskette remaining. This procedure can already be on the remaining diskette, copied to it from the diskette being removed or created on it by a program.

PROC has to be used rather than SUBPROC so that the procedure file on the diskette to be removed is closed.

When the original diskette is reinserted, the procedure on the diskette that remained issues a PROC command back to a procedure on the reinserted diskette.

For example, the following sequence does the back up steps for the diskette in slot 2. The initially active procedure (MENU) is on the diskette in slot 1:

```
Slot 1                              Slot 2

MENU procedure
 LOAD MENU.PROGRAM
 RUN
  CHAINs to PROC=COPY
  (if copy option selected)

                                    COPY procedure
                                      prompts to insert backup
(Backup inserted)
                                    LINK COPY
                                      copy parameters
                                    ENDLINK
                                      prompts to reinsert original
(Original inserted)
                                    PROC MENU

MENU procedure
 LOAD MENU.PROGRAM
 RUN
```

In this example the diskette containing the active procedure remains inserted until that procedure is closed by issuing a PROC command to the other diskette.

If both diskettes need to be removed, the chain of procedures has to be arranged so that before each diskette is be removed a PROC command activates a procedure on a diskette that remains mounted. Both diskettes cannot be removed at the same time.

CREATING PROCEDURES FROM A PROGRAM

Frequently the steps of a procedure or file names used in a procedure depend upon the specific sequence of activities the application user selects. Procedures can be created by a program by putting the desired commands into a display file. For example, the following program steps create a procedure that does a SORT using a sort control file name contained in the variable NAME$:

```
00020 OPEN #1:"NAME=srt.proc/vol",DISPLAY,OUTPUT
00030 RESTORE #1:
      .
      .
      .
00100 PRINT #1:"SORT "&NAME$
00110 PRINT #1:"PROC MENU"
00120 CLOSE #1:
00130 CHAIN "PROC=SRT.PROC/VOL"
```

Line 20 opens the procedure file as a display file for output. The file is assumed to exist.

Line 30 resets the record pointer to the beginning of the file.

Line 100 Puts the SORT command into the file as the first procedure step. The entry is the word SORT followed by a blank and the contents of the variable NAME$, which contains the name of the sort control file to be used by SORT.

Line 110 Puts a PROC command in the file so that when the sort completes, the MENU procedure is executed, thus closing the SRT.PROC file so that it can be rewritten (if necessary) for the next operation.

Line 120 closes the newly created procedure file.

Line 130 chains to the newly created procedure (which ends the execution of this program).

There are some rules that must be understood about updating or creating procedure files from a program:

1. An active procedure file cannot be rewritten. A procedure file remains active until another PROC command is issued, or the last step in the procedure completes and an end-of-file occurs attempting to get the next procedure step.

   This is the reason the "PROC MENU" step was added to the procedure created above in line 110.

2. Individual records in a procedure file cannot be updated because it is a display file (Type 05). Therefore, the entire procedure must be rewritten any time it is changed.

PREVENT AN EXTRA LINE FROM PRINTING AFTER NEWPAGE

If extra lines creep into a report, it may be caused by the PRINT #255:NEWPAGE statement. This can be eliminated by following the statement with two semicolons:

```
PRINT #255:NEWPAGE;;
```

HOW TO PRINT 198 CHARACTERS/LINE AT 15 CPI

When printing at 15 characters per inch (CPI), the computer can print 198 characters on a line (132*1.5). However, the default line length for the printer (device #255) is 132 characters. This can be changed by opening the printer and setting a new record length:

```
OPEN #255:"NAME=//10,RECL=198",DISPLAY,OUTPUT
```

HOW TO CHANGE THE PAGE OVERFLOW COUNT

When using forms that are a different length than the standard 66 lines, or when printing at a different number of lines per inch, the page overflow count (normally 60 lines) needs to be changed. This can be done by opening the printer and setting a new page overflow count:

```
OPEN #255:"NAME=//10,PAGEOFLOW=80",DISPLAY,OUTPUT
```

NOTE: When the printer is turned OFF or the Error Reset key is pressed, the printer will reset to the normal printer defaults. Therefore, when you change the printer defaults (such as horizontal density, vertical density, lines per inch, quality print), you should have the BASIC statements that issue those changes as part of your error handling routines.

ERROR 0802 USING PIC(Z.ZZ) FORMAT SPECIFICATION

Error 0802 occurs when running a program that has a FORM statement with a PIC specification with zero suppress Zs after the decimal point.

The Datamaster does not support this type of specification. This is documented in the BASIC LANGUAGE REFERENCE manual (SA34-0109).

ERROR 0726 OR OTHER MAPPING ERRORS

Error 0726 or other mapping errors may occur if the replication factor for a FORM statement specification is greater than 255. This is a design limit of the computer and is documented in the BASIC LANGUAGE REFERENCE manual (SA34-0109).

This problem can be avoided by "breaking" the specifications into pieces smaller than 255. For example:

If the FORM statement wanted is:

```
00100 FORM 400*PD 5,C 10
```

Then use:

```
00100 FORM 200*PD 5,200*PD 5,C 10
```

SIGN DOES NOT REPLACE THE COMMA IN PIC

A floating minus (-) or plus (+) sign does not replace the comma (,) in a FORM statement PIC specification. For example:

```
00100 LET X=-345.67
00110 PRINT #255,USING 120:X
00120 FORM PIC(--,---.##)
```

The computer prints:

```
 - 345.67
```

instead of:

```
-345.67
```

If the position of the sign is critical to the report or form being printed, do not include commas in the PIC specification.

PREVENT THE DISPLAY SCROLLING WHEN SOUNDING THE ALARM

When the alarm is sounded using the PRINT BELL statement during full-screen-processing, the display scrolls up one line unless the statement is followed by two semicolons:

```
PRINT BELL;;
```

INCORRECT RESULTS FROM INPUT FIELDS

If the cursor positioning keys (the 'arrow' keys in the lower left corner of the keyboard) are used to move from field to field, incorrect data may be transferred to the variables of the INPUT FIELDS statement.

This problem can be avoided by always using the Field Advance, Field Backspace, Field Exit, New Line, Field Plus or Field Minus keys to move from field to field.

The INPUT FIELDS specification string can also contain attributes that automatically do a field advance (A) or enter (E) thus minimizing the use of field or cursor movement keys (see the BASIC LANGUAGE REFERENCE manual (SA34-0109), Full-Screen Processing section).

ERROR 6126 USING SUBSTRING IN INPUT FIELDS SPECIFICATION

Error 6126 may occur if the INPUT FIELDS specification contains a substringed variable. For example:

```
00100 INPUT FIELDS "2,"&COL$(2:3)&",C 8,N,N":X$
```

This error can be avoided by assigning the substringed variable to another variable and using that variable in the INPUT FIELDS specification. For example:

```
00090 LET CC$=COL$(2:3)
00100 INPUT FIELDS "2,"&CC$&",C 8,N,N":X$
```

HOW TO POSITION CURSOR AT FIELD CAUSING AN ERROR

When an application uses full-screen processing to enter numerous items, it is nice to be able to position the cursor at the exact field where an error occurred. This assists the user in quickly correcting the error.

Doing this requires two things: identifying which of several fields caused the error and changing that field's attributes to place the cursor there. The following example shows how this can be done and also highlights the field in error:

```
00020 DIM FMT$(3),IN$(3)
00030 LET FMT$(1)="02,02,N 5,UH ,N"
00040 LET FMT$(2)="03,02,N 5,UH ,N"
00050 LET FMT$(3)="04,02,N 5,UH ,N"
      .
      .
      .
00100 LET C=1
00110 INPT: INPUT FIELDS MAT FMT$:MAT A$ CONV CERR
00120 LET FMT$(C)(11:13)="UH "
00130 REM NO ERROR OCCURRED
      .
      .
      .
00200 CERR: LET FMT$(C)(11:13)="UH "
00210 LET C=CNT+1
00220 LET FMT$(C)(11:13)="RHC"
00230 PRINT BELL;;
00240 GOTO INPT
```

Line 20 dimensions the format specification and input variable arrays.

Lines 30-50 set initial format specifications.

Line 100 sets a field attribute to reset if no error occurs.

Line 110 does the full screen input of the 3 fields. If a conversion error occurs the program branches to the line labeled CERR.

Line 120 clears the cursor and reverse image attributes from the last field in which an error occurred (if any) so that it will not affect the next use of this INPUT FIELDS.

Line 200 clears the cursor and reverse image attributes from the last field in which an error occurred (if any).

Line 210 sets C to the field number that caused the error (CNT is the last successful field processed).

Line 220 sets the cursor and reverse image attributes for the field that caused the error.

Line 230 sounds the alarm to notify the user that an error has occurred.

Line 240 returns to the INPUT FIELDS to retry the input. This time the field that caused the error is in reverse image and the cursor is positioned at that field.

(This page intentionally left blank.)

PROBLEM WITH RELEASE CHANGE ON 5247 DISK

Some disk users operated with BRADS III, release 2.0 on the 5247 Disk Unit prior to receiving the current BRADS III, release 2.3 diskettes. By using the older release, 2.0, they may have inadvertently placed a special BRADS file on one or more of the BRADS diskettes used in the application. The result could be a confusing set of error messages, such as: "FILE CANNOT BE FOUND" and "DEFINITION NOT IN DIRECTORY." When this occurs, BRADS is using one of the old and possibly inaccurate special files.

To correct this problem, perform these steps:

Key: PRINT WSID$ and press the Enter key.

The response will be the two-digit work station ID such as 11, 21, 31, or 41.

Use the DIR command to print the directory of every diskette used in the application.

Search the listings for the name:

BRFUSExx where xx is the same as the WSID$ printed above.

Use the FREE command to free these files wherever they are found.

NOTE: There should be only one BRFUSExx file present in the system and it should be located on the BRADS3 volume, whether that is a diskette or one of the disk volumes.

Retry the BRADS application. If BRADS is unable to find a proper BRFUSExx file, Key: CMD/ERROR RESET and BRADS creates a new BRFUSExx file.

BRADS III EXAMPLES

BRADS III Examples are included in the BRADS material as a guideline for writing BRADS reports to interface with your current BMAS applications. The techniques used in BRADS Examples can be helpful in building reports to interface with other accounting applications.

The intent of the "Examples" is to illustrate the use of BRADS to produce a variety of additional reports to supplement your accounting application reports. They are NOT tested, ready to run programs, to use as guidelines to assist you in generating your own BRADS reports.

BRADS III, mod 3, release 2.3, is the most current level of BRADS. The Examples sent with the new release of BRADS are the original level without the various PTFs applied as Examples are not designed to be running code, but merely educational guidelines. Therefore, if you have modified your BRADS Examples or applied various PTFs, you will not want to copy the new Examples over your existing Examples.

TO DETERMINE THE RELEASE OF BRADS

The BRADS III program diskette and the Spread diskette contain history files which contain information regarding the release and modification level of the BRADS programs. To access this information on the BRADS III operations diskette,

Key:

```
LOAD PTF.HIST,DATA        and press the ENTER key
```

The response will be:

```
00010:BRADS III, Release xx Mod Level yy PTF xx date
      83/10/28
```

where xx is the release number and yy is the modification level.

To access the information on the Spread diskette,

Key:

```
LOAD PTF.HIST2,DATA       and press the ENTER key
```

If the modification level of the BRADS III diskette is less than 02, see your marketing representative for a new release of BRADS.

RE-INSTALLING BRADS III WITH RELEASE 2.3

If BRADS III has been in use for a while, it probably has many definitions in the BRADS directory DIR. There may also be other BRADS directories on the BRADS operations diskette or stored reports there. How is all this information saved when Release 2.3 of BRADS III is installed? Just follow these steps:

1. Install BRADS III Release 2.3 using the directions in LEARNING BRADS III BOOK 1. Be sure to use a blank diskette when creating the new operations diskette. Don't use your current operations diskette.

2. Now that a new operations diskette has been created, copy all the definitions from the DIR directory on the old operations diskette to the DIR directory on the new operations diskette. Use the 'Copy Directory' command to do this. The entries on screen 8103A should look something like this:

```
FROM DIRECTORY    DIR//1        ---Old diskette in drive 1
TO DIRECTORY      DIR//2        ---New diskette in drive 2
DEFINITION NAME   *             ---* means all definitions
ON NAME MATCH     R
```

   Notice that the drive number is specified. This is so that BRADS can tell which directory is which.

3. Any files (BRADS directories, reports, etc.) that are on the old BRADS operations diskette should be copied to the new operations diskette. Use the COPY customer support function (Choice 2) to do this. If it is unclear which files should be copied, print directories (DIR 1,PRINT) for each diskette and compare them. Copy any files that are on the old diskette and not on the new diskette.

When you have completed these steps, you can use the new operations diskette as you did the old one.

RE-INSTALLING BRADS III WITH RELEASE 2.3 TO A 5247 DISK

1. Type RENAME DIR/BRADS3/5,DIR.TEMP and press ENTER

2. Type RENAME DIR.I/BRADS3/5,DIR.I.TEMP and press ENTER

   NOTE: This will hold your current BRADS directory and its index file in temporary files so that they are not destroyed during the installation of the new release of BRADS III.

3. Insert the BRADS diskette number 1 of 5 into drive 1

4. Type PROC INSTALL.BRAD.DISK and press ENTER

5. When the CPU displays screen 42-169, press the ERROR RESET key and then press CMD 9

6. When the next BRADS installation screen is displayed press the ENTER key to continue

7. When the BRADS installation has completed and you have returned to the READY INPUT screen, proceed to step 8

8. Type: FREE DIR/BRADS3/5 and press ENTER.
   Type: FREE DIR.I/BRADS3/5 and press ENTER.
   Type: RENAME DIR.TEMP/BRADS3/5,DIR and press ENTER.
   Type: RENAME DIR.I.TEMP/BRADS3/5,DIR.I and press ENTER.

Your system will now be updated to the newest release of BRADS III and you are ready to continue your normal operations of BRADS III.

SYSTEM/34 BRADS FILE QUERY

Running File Query on large files results in a slower operation if the report is printed immediately after the program is generated.

Use the following technique to interrupt and save the File Query program after it is generated. Then, run the File Query program using the Run Report option on the BRADS menu, or load and run it from BASIC.

1. First "Generating FILE QUERY Running Code" appears on the lower left of the display

2. When the code is generated the following is displayed

```
File Query Running
Press ATTN to Interrupt
```

3. Press the ATTN key

4. When the "Inquiry Options" are displayed, select option 4 - Set Inquiry Condition for Program

5. When the "File Query Interrupted" screen is displayed, Type a CMD 6 - Save Program

6. When screen 8122E - BRADS - FILE QUERY/SAVE QUERY PROGRAM is displayed, enter a Query Program Name and enter 'N' in response to the "Return to File Query after Save" prompt and press the Enter key. Note that the program name that is entered should be different from any other file name on the disk.

PROCESSING DUPLICATE-KEY INDEX FILES IN BRADS

This technique is used to process large files in which several records have the same key. For example, an open receivables file in an accounts receivable application has several invoice or credit memo records for each customer. An index file built for this file using just the customer number has duplicates.

This type of file can be processed in a BRADS report as follows:

1. Assume F$(1) is the key field in the file definition of the transaction file that contains duplicates

   Generate an index on this field and allow duplicate keys

2. Assume Screen 1 will be displayed allowing for entry of the desired customer number which is put into F$(33)

3. Enter Specify Report and select File Specifications

   Enter index definition name

   Input method 'D' for demand

4. Select Calculations

   Display screen 1 and provide for entry of the desired key value into F$(33). Code first access to the file through the index, then code subsequent reads sequentially. For example:

```
30000  IF FNSCREEN(1,1,0,1)=9 THEN GOTO BEND
30010  ON FNREAD(F1,33) THEN GOTO 30020,BERRCALC,30000
30020  A=FNPRINT(P1,0,1)
30030  ON FNREAD(F1,SEQ) THEN GOTO 30040,30000,BERRCALC
30040  IF F$(33)><F$(1) THEN GOTO 30000
30050  A=FNPRINT(P1,0,1)
30060  GOTO 30030
```

Line 30000 displays a screen that prompts the user to enter the number (key) of the record to be processed.

Line 30010 reads file 1 using the value entered into F$(33). If the key is found, continue to line 30020. If EOF is reached (an error on an indexed read), go to the BRADS error routine BERRCALC. If no record is found, go back to the screen to allow entry of another key value.

Line 30020 processes the record. In this example, each record selected will be printed. Any other processing of the record is done at this point.

Line 30030 reads the subsequent records sequentially.

Line 30040 compares the key value entered into F$(33) to the key field F$(1) from the file to make sure that the record read has the same key as the first record found. If it is not equal, then go to the screen so another key value or end of job can be entered.

Line 30050 processes the record.

Line 30060 goes to the sequential read to get the next record.

NOTE: Use of the BRADS functions FNREAD, FNPRINT, FNSCREEN may be found in the BRADS III REFERENCE manual (SB30-2538).

BRADS REPORTS BUILT WITH SPECIFY REPORT

Any report written with the Specify Report function of the BRADS application is a stand-alone, BASIC program which may be executed independently of BRADS. Once created, you may use CMDs: LOAD, LIST, LISTP, RUN, RUN TRACE, REPLACE.

If changes need to be made to the user-written calculations, the report program may be LOADed, modified, and then REPLACEd. Any changes that are made other than to the calculations, for example, file specifications, print formats etc., must be made through the Specify Report option of the BRADS menu and the report must be rebuilt.

If changes to the BRADS code are necessary, the program must be LOADed, modified, and REPLACEd. Should the program be rebuilt using the Specify Report option of BRADS, all previous user modifications to the generated code are lost.

NOTE: When specifying Free-Form reports in BRADS which require user-written calculations, specify all options required except calculations and build the report. Then LOAD the program into memory, add the needed program statements, and REPLACE it.

NOTE: A report program generated and saved by the Query option of BRADS is also a stand-alone, BASIC program. It may be manipulated as stated above, executed through the run report option, or RUN independently of BRADS.

CREATING 3-ACROSS MAILING LABELS IN SPECIFY REPORT

To create three-across labels in specify report which will use a different name for each label, such as:

```
Joe Smith           Mary Jones          Rich McAndrews
124 Elm             124 Maple           111 Jackson
Suite 100           Niles, IL  60014    Suite 300
Eldon, IL  60012                        Waco, TX  78021
```

all character fields should be defined as variable to eliminate extraneous blanks in data fields. The input method for the data file should be specified as 'S' for sequential. In the report format, the field design should consist of rows of pound signs for each address line in three columns. Each line should use a unique field reference that is not used in the data file with each column starting with a multiple of ten.

```
f$(11)-------       f$(21)-------       f$(31)-------
f$(12)-------       f$(22)-------       f$(32)-------
f$(13)-------       f$(23)-------       f$(33)-------
f$(14)-------       f$(24)-------       f$(34)-------
```

Calculations are required in the BCYCLE and BREAD area of the report.

```
05000 MAT F$=("")
05010 FOR COL=10 TO 30 STEP 10
11000 LET F$(COL+1)=F$(1)&" "&F$(2)
11010 LET F$(COL+2)=F$(3)
11020 LET F$(COL+3)=F$(4)
11030 LET F$(COL+4)=F$(5)&", "&F$(6)&" "&F$(7)
11040 IF F$(4)><"" THEN GOTO 11070
11050 F$(COL+3)=F$(COL+4)
11060 F$(COL+4)=""
11070 NEXT COL
21000 GOSUB BPDETAIL
30000 GOTO BCYCLE
```

Line 05000 clear array F$ to avoid duplicate labels on last row

Line 05010 10's = column 1, 20's = column 2, 30's = column 3

Line 11000 current column line 1 = first name, a blank space, and last name

Line 11010 current column line 2 = first address line

Line 11020 current column line 4 = second address line

Line 11030 concatenate city, a comma and a blank, state, a blank, and the zip code

Line 11040 if the second address line is not blank, do not compress the address lines

Line 11050 move line 4 up to line 3

Line 11060 blank out the 4th line

Line 11070 next column in the loop

Line 21000 the three columns are now complete, print the detail lines

Line 30000 labels printed, start the cycle again

BRADS SPREAD INPUT FILES

BRADS Spread users should review the "Input Files" and "File Opcode" sections of the BRADS REFERENCE (SPREAD) manual (SB30-2540). It is faster to use BRADS maintained input files than entering the data into the Spread Specifications file as shown in LEARNING BRADS Book 3. Keying original data, changes, 'actual' monthly figures and finding records quickly is easier and faster once input file usage is understood.

Use the input file examples described in the sample spread sheets in the "Consolidations" and "Plan-vs-Actual" pages of the BRADS REFERENCE (SPREAD) manual to gain a greater understanding of BRADS input file concepts.

Some input file highlights:

1. When designing BRADS input files for spread reports insure that in addition to the spread data these 3 identification and control fields are included in the file definition:

   Row Name(limited to 10 characters)
   Row Description(ranging from 10 to 30 characters)
   A 'P' Code(3 characters).

   These fields will be used for both selection and printing purposes. If files already exist without these fields, it might be possible to use record number for selection of specific rows of data.

2. A model file definition, such as 'monthly' shown in BRADS LEARNING Book 3, can be used to simplify preparation of spread sheets. That definition includes 13 numeric data fields and the 3 identification and control data fields described in 1 above. This

definition could represent 12 individual monthly and 1 annual total values for a typical accounting record. Once this definition is placed in the BRADS directory, it can be used as a model in Define File to build a definition for an actual data file. A model definition of a file having 12 months actual and 12 months budget data is shown as 'VARIANCE.DATA' on page D-24, in the BRADS REFERENCE (SPREAD) manual, (SB30-0540).

3. Substituting data files is another time-saver as described on page S-83 under Specify Spread in the BRADS REFERENCE (SPREAD) manual, (SB30-0540). The file description 'monthly' is used with the actual data being retrieved from other files. Review the various parameters (file,dir,y,n) of these examples to understand the built-in flexibility of using various VOLIDs and DIRs.

4. In order to use input files in spread it is necessary to indicate 'Y' for the input file option on Specify Spread screen 8133C and specify at least one File Opcode calculation to select information from the input file.

5. Some suggestions for using the File Opcode:

   a. Place File Opcodes at the beginning of the calculation specifications

   b. Place File Opcode calculations in the input file order rather than the order that spread processes them (such as, information from Record 1 first (or File 1 first, if multiple files are used))

   c. Extra Formatting or Run Report can be used to re-arrange calculated data rows when printing if required.

   d. Individual rows or groups of rows can be retrieved by row name (if available) or by record number.

   e. The Rowseq Opcode can be used to specify the sequence of rows to be printed or to select one of a group of rows to use in a spread calculation.

   f. Consolidation of several files (of similar format) is easy.

   g. If a row is used in a calculation specification and a subsequent File Opcode selects a row by the same name from a file, the two rows will be added together. This design permits the consolidation of files.

(This page intentionally left blank.)

IBM®

G360-1000-0