

Loading and Managing Extended BASIC

NOTICE

DATA GENERAL CORPORATION (DGC) HAS PREPARED THIS DOCUMENT FOR USE BY DGC PERSONNEL, LICENSEES, AND CUSTOMERS. THE INFORMATION CONTAINED HEREIN IS THE PROPERTY OF DGC; AND THE CONTENTS OF THIS MANUAL SHALL NOT BE REPRODUCED IN WHOLE OR IN PART NOR USED OTHER THAN AS ALLOWED IN THE DGC LICENSE AGREEMENT.

DGC reserves the right to make changes in specifications and other information contained in this document without prior notice, and the reader should in all cases consult DGC to determine whether any such changes have been made.

THE TERMS AND CONDITIONS GOVERNING THE SALE OF DGC HARDWARE PRODUCTS AND THE LICENSING OF DGC SOFTWARE CONSIST SOLELY OF THOSE SET FORTH IN THE WRITTEN CONTRACTS BETWEEN DGC AND ITS CUSTOMERS. NO REPRESENTATION OR OTHER AFFIRMATION OF FACT CONTAINED IN THIS DOCUMENT INCLUDING BUT NOT LIMITED TO STATEMENTS REGARDING CAPACITY, RESPONSE-TIME PERFORMANCE, SUITABILITY FOR USE OR PERFORMANCE OF PRODUCTS DESCRIBED HEREIN SHALL BE DEEMED TO BE A WARRANTY BY DGC FOR ANY PURPOSE, OR GIVE RISE TO ANY LIABILITY OF DGC WHATSOEVER.

This software is made available solely pursuant to the terms of a DGC license agreement which governs its use.

CEO, DASHER, DATAPREP, ECLIPSE, ENTERPRISE, INFOS, microNOVA, NOVA, PROXI, SUPERNOVA, PRESENT, ECLIPSE MV/4000, ECLIPSE MV/6000, ECLIPSE MV/8000, TRENDVIEW, SWAT, GENAP, and MANAP are U.S. registered trademarks of Data General Corporation, and **AZ-TEXT, DG/L, DG/GATE, DG/XAP, ECLIPSE MV/10000, GW/4000, GDC/1000, REV-UP, UNX/VS, XODIAC, DEFINE, SLATE, microECLIPSE, DESKTOP GENERATION, BusiPEN, BusiGEN** and **BusiTEXT** are U.S. trademarks of Data General Corporation.

Loading and Managing
Extended BASIC
093-000119

Revision History:

Previous title: Extended BASIC System
Manager's Guide
Original Release: February 1975
First revision: June 1975
Second revision: October 1977
Third revision: June 1979
Fourth revision: October 1983

Effective with:

Extended BASIC 1.60 (AOS and AOS/VS)
Extended BASIC 5.60 (RDOS, DOS, and
DG/RDOS)

Loading and Managing Extended BASIC

093-000119-04

For the latest enhancements, cautions, documentation changes, and other information on this product, please see the Release Notice (085-series) supplied with the software.

Ordering No. 093-000119
©Data General Corporation, 1975, 1977, 1979, 1983 All Rights Reserved
Printed in the United States of America
Revision 04, October 1983
Licensed Material - Property of Data General Corporation

Summary of Changes

Loading and Managing Extended BASIC is a revision of the *Extended BASIC System Manager's Guide*. The manual has been reorganized to promote ease of reference. The most significant changes are:

- Former chapter 1 (Introduction to BASIC System Management) has been eliminated. The subject of old chapter 1, performance considerations, is discussed in chapter 3 of the current manual.
- Former chapter 5 (Extend BASIC Privileged Commands and Statements) has been eliminated. The "privileged" commands and statements are now documented in chapter 3 of the *Extended BASIC Reference*.
- Discussion of SOS Extended BASIC has been eliminated from chapter 1 of the current manual.
- Instructions for loading Extended BASIC files from the release tape or diskette have been added to chapter 1.
- Appendix B, which used to be a summary of the privileged commands and statements only, is now a summary of all BASIC commands and statements but with the privileged commands and statements duly noted.

The new material and the technical revisions and corrections to previously documented material are marked with revision bars (|) in the margins.

Contents

Preface

Chapter 1 - Extended BASIC Installation

Installing Extended BASIC on RDOS and DOS	1-1
RDOS and DOS System Generation	1-1
Loading BASIC Modules	1-2
BASIC System Generation Dialog for RDOS and DOS	1-3
Creating BASIC Disk Directories and Files	1-8
Library Directory: BASIC.DR	1-9
User Directories	1-9
BASIC.ID File	1-9
Logon Programs	1-9
BASIC.AF File	1-10
Installing Extended BASIC on RTOS	1-10
RTOS System Generation	1-11
RTOSGEN	1-11
BASIC System Generation Dialog Under RTOS	1-12
BASIC Commands, Statements, and Functions on RTOS	1-15
Installing Extended BASIC on AOS and AOS/VS	1-15
Loading BASIC Modules	1-16
Calling Assembly Language Subroutines	1-16
Changing the Overlay Configuration	1-17
Managing the User Environment	1-17
Invoking Extended BASIC	1-17
Shared Files and Record Locking	1-17
Testing Extended BASIC	1-18

Chapter 2 - Start-up, Shutdown, and Crash Procedures on RDOS and DOS

Cold Starting BASIC	2-1
System Preparation	2-1
Invoking the BASIC System	2-1
Stopping the BASIC System	2-1
Recovery from System Crashes	2-2
Updating the Records of a File	2-2
Auto Restart Procedure	2-3
System Backup Procedures	2-3

Chapter 3 - Extended BASIC System Operation on RDOS and DOS

System Operator Responsibility	3-1
Communicating with Users	3-1
Changing Directories	3-1
System Console Breaks	3-1
Device Independence	3-1
System Manager Programs	3-2
System Performance Considerations	3-9
System Security Considerations	3-10
Controlling Access to the System	3-10
Directory Sharing	3-10
Monitoring the System	3-11
Protecting Programs and Data Files	3-11
Batch Operation	3-11

Appendix A - Initialization Error Messages

Appendix B - Statement, Command, and Function Summary

Appendix C - Sample RDOS BSG Dialog

Appendix D - ASCII Character Set

Index

Tables

Table

A-1	Initialization Errors	A-2
-----	-----------------------------	-----

Illustrations

Figure

1-1	Disk File Space Before and After Partitioning	1-8
3-1	Program EDITID.QA (continues)	3-3
3-1	Program EDITID.QA (concluded)	3-4
3-2	Results of Program EDITID.QA	3-4
3-3	Program USERS.QA	3-5
3-4	Results of Program USERS.QA	3-6
3-5	Program FREEID.QA	3-7
3-6	Results of Program FREEID.QA	3-8
3-7	Program KILLID.QA	3-9
3-8	Results of Program KILLID.QA	3-9
3-8	Memory Allocation, in Words, in an RDOS System	3-10
3-10	Sample BATCH BASIC Program	3-12

Preface

This manual explains how to install Extended BASIC and discusses topics that a system manager or operator needs to consider.

Chapter 1 explains how to install Extended BASIC from the released media, under each of the available operating systems: RDOS, DOS, RTOS, AOS, and AOS/VS.

Chapter 2 is relevant only to RDOS users. It explains how to start up and halt Extended BASIC, how to back up the system, and what to do if the system crashes.

Chapter 3, also relevant only to RDOS users, describes daily operating procedures, performance considerations, and system security.

Appendix A lists and briefly explains Extended BASIC initialization error messages.

Appendix B lists alphabetically and summarizes briefly the Extended BASIC commands, statements, and functions. The appendix also indicates the RDOS and DOS privileged commands and statements, which can be executed only from the master terminal.

Appendix C shows a sample dialog for Extended BASIC system generation (BSG) on RDOS.

Appendix D contains an ASCII character chart.

Related Documents

Users of this manual should be familiar with the CLI or reference manual for their particular operating systems, and the *Extended BASIC Reference* (093-000065). The following documents offer further information:

093-000075	Real Time Disk Operating System Reference Manual (RDOS)
093-000201	Disk Operating System (DOS) Reference Manual
093-000056	Real Time Operating System Reference Manual
093-000109	RDOS/DOS Command Line Interpreter User's Manual
093-000188	How to Load and Generate Your RDOS System
093-000222	How to Generate Your DOS System
093-000087	Batch User's Manual

093-000122	Command line Interpreter User's Manual (AOS and AOS/VS)
093-000194	Advanced Operating System (AOS) Operator's Guide
093-000244	Advanced Operating System/Virtual Storage (AOS/VS) Operator's Guide
093-000217	Managing AOS
093-000243	Managing AOS/VS

Reader, Please Note

The following are conventions for command formats in this manual:

COMMAND required *[option]*...

Where	Means
COMMAND	Enter the command (or its accepted abbreviation) as shown.
required	Enter the given argument, such as a filename. In the following instance enter one of the arguments but not the braces, which merely set off the choices: argument1 argument2
<i>[option]</i>	You have the option of entering the given argument. Don't enter the brackets, which merely set off the optional argument.
...	You may repeat the preceding entry or entries. The explanation tells you what you may repeat.

The following formats are used in this manual:

All numbers are decimal unless shown otherwise; e.g., 35g.

In examples of system interactions, the manual uses

THIS TYPEFACE TO SHOW YOUR ENTRY
ITALIC TYPEFACE FOR SYSTEM QUERIES AND RESPONSES

) is the AOS and AOS/VS CLI prompt.

R is the RDOS/DOS CLI prompt.

* is the Extended BASIC prompt.

The word *terminal* means a CRT display terminal, or equivalent interactive device.

End of Preface

Chapter 1

Extended BASIC Installation

This chapter is divided into three main sections:

- Installing Extended BASIC on RDOS and DOS
- Installing Extended BASIC on RTOS
- Installing Extended BASIC on AOS and AOS/VS

Readers should go to the appropriate sections, depending on the versions of Extended BASIC they wish to install.

Installing Extended BASIC on RDOS and DOS

The steps for installing Extended BASIC on RDOS and DOS systems are similar:

1. Tailor the operating system (RDOS or DOS) to Extended BASIC requirements.
2. Load the Extended BASIC modules from the release tape or diskettes.
3. Generate Extended BASIC.
4. Create Extended BASIC disk directories and files.

RDOS and DOS System Generation

The system generation program (SYSGEN, BSYSGEN, or NSYSGEN — depending on your hardware) tailors an operating system for your particular environment, based on your responses to prompts from the program. The system-generation dialog is not explained in detail here. However, the prompts that are critical to the requirements of Extended BASIC are listed below in the order in which they appear in the dialog.

Maximum number of channels BACKGROUND/BACKGROUND will use

In an unmapped RDOS system and all DOS systems, channels are taken from user memory, and this question is not asked. In a mapped system channels are taken from system memory, and the space must be pre-allocated. Typically, RDOS users generate fifteen channels or fewer; BASIC, however, may require one hundred or more. The number of channels specified should be at least:

- 3 for system support
- 2 more for multiuser systems

- 1 more if there is to be swapping
- 1 more if there is to be accounting
- 2 more for each user terminal
- n more, where n is the maximum number of files that users may open simultaneously, as declared in the Extended BASIC system generation dialog. (See prompt 17 below, *Maximum files to be open at one time.*)

Number of stacks

This is the number of simultaneous active system calls. Single-user BASIC requires at least two stacks; multiuser BASIC requires at least three. These are hard limits, not recommendations. More than five stacks per ground are rarely useful.

Number of extra cells

Cells are used to keep track of inactive system calls when stacks are exhausted and for special calls (such as multiplexor I/O) which do not use stacks. Cells are also used to buffer multiplexor input beginning in RDOS Rev. 6.52. One or two extra cells per multiplexor line should be sufficient.

Number of extra buffers required

You get two buffers per stack, with a minimum of six buffers per system. In small systems, additional stacks are more worthwhile than extra buffers. Specify zero unless (1) you already have more than five stacks and the tuning report indicates a shortage of buffers, or (2) you are running INFOS. In the latter situation your database design determines the need for extra buffers.

In larger systems, many extra buffers may be useful, especially if space for virtual overlays and extended memory swapping is extensive.

Maximum number of sub-directories/sub-partitions accessible at one time

BASIC usually needs one directory per user, plus one. RDOS users frequently specify one or two; BASIC, however, may require thirty or more directories. Remember that in multiuser systems each user at any given time can access a different directory. Your answer to this

question should be large enough to accommodate this situation.

RTC?

Multiuser BASIC requires a real time clock. It adapts automatically to any clock frequency (including microNOVA internal), but usually the lowest speed available is best.

QTY?

ALM?

ULM?

Multiuser BASIC requires one of the above. The system does not support a mixture of hardware.

Use default ALM/QTY interrupt characters?

BASIC requires ESC as an interrupt character. Answer 0 to use the default, and specify decimal code 27 (ESC) as the first character and 128 (none), or some other preferred character, for the second.

Core dump facility?

Core dumps may be required to diagnose problems, but line printer dumps are vastly inferior to other media. If possible, specify 2, for magnetic tape; 3, for diskette, will do in emergencies.

Loading BASIC Modules

BASIC is released as a group of library and relocatable binary modules on magnetic tape or diskette, depending on the type of computer system you have. Below are instructions for loading the released files onto your system. After loading, store the release tape or diskettes in a safe place in case you should accidentally erase or destroy your BASIC system.

We recommend that you load the Extended BASIC system generation (BSG) files into a subdirectory. Doing so makes it easier to delete the numerous BSG files after the BASIC system is complete. The following instructions assume that you load the BSG files into subdirectory BASICGEN.DR.

The steps for loading the Extended BASIC release tape or diskette onto your system follow. Note that x stands for the number of the drive containing the release tape or diskette.

1. Create the BASICHELP.DR directory in your master directory. If a BASICHELP.DR directory already exists in your master directory, first delete it and then re-create it. The command is:

CDIR BASICHELP

2. Load the BASICHELP files from the release tape or diskette into the BASICHELP.DR directory. For loading from tape, the commands are:

INIT MTx
DIR BASICHELP
LOAD/V/R/A MTx:0 -.HF

For loading from diskette, the commands are:

INIT BASICHELP
DIR DPx
MOVE/V/R/A BASICHELP -.HF

3. Create a directory for use during Extended BASIC system generation (BSG). For example,

CDIR BASICGEN.DR

4. Load the Extended BASIC modules from the release tape or diskette into the BSG directory. The commands for loading from tape are:

INIT MTx
DIR BASICGEN
LOAD/V/R/A MTx:0 -.HF/N

The /N template prevents the loading of the help files, which have already been loaded in step 2, above.

For loading from diskette, the commands are:

INIT BASICHELP
DIR DPx
MOVE/V/R/A BASICGEN -.HF/N

5. Link or rename the system library (SYS.LB) to one of the following, depending on the type of system the generated Extended BASIC is to run on:

USYS.LB	Unmapped NOVA system library (DOS or RDOS)
MSYS.LB	Mapped NOVA system library (RDOS)
BSYS.LB	Unmapped ECLIPSE system library (RDOS)
ASYS.LB	Mapped ECLIPSE system library (RDOS)

You can generate an Extended BASIC for any of these target machines, provided you have on your system an appropriately named system library. Note that if both NOVA and ECLIPSE target machines are to be supported, you must obtain and load both NOVA and ECLIPSE models of Extended BASIC.

6. Set up links in the BSG subdirectory to wherever the appropriate SYS.LB, RLDR.SV, RLDR.OL, and PATCH.SV files reside on your disk.

You are now ready to generate Extended BASIC, discussed next. After completing Extended BASIC generation, check the appropriate release notice for patches that may be necessary.

BASIC System Generation Dialog for RDOS and DOS

To generate the BASIC system, enter the following command:

BSG

This command executes the program *BSG.SV*, which displays the following identification message, and then asks for your instructions for building your configuration of Extended BASIC. (Note that *xx.xx* represents the revision level of Extended BASIC, and *mm/dd/yy*, the release date.)

Extended BASIC Revision xx.xx mm/dd/yy System Generation

Valid answers to prompts appear within parentheses; the default value is given first. Terminate each response (except that to question 23) with a carriage return. You may select the default response by simply pressing the carriage return, with nothing before it. For example, if you press carriage return in response to question 2,

Object operating system (RDOS,RTOS,DOS)?

The program uses the default value RDOS in generating BASIC. If you wish to specify one of the alternatives, enter its name as it appears in the prompt, and press the carriage return.

The following queries and their responses are numbered for easier reference. BSG does not display these numbers.

1. *Object computer type (NOVA,ECLIPSE,ORIGINAL)?*

Identify the type of computer on which the BASIC system is to be run. ORIGINAL refers to the original NOVA processor, which had a slightly different multiply/divide instruction than subsequent NOVA and ECLIPSE computers. The NOVA response covers 1200 series, 800 series, NOVA 2, NOVA 3, and NOVA 4 computers, as well as microNOVA, MP100, and MP200 configurations. The ECLIPSE response covers all models except the M600, which requires AOS BASIC.

2. *Object operating system (RDOS,RTOS,DOS)?*

Respond with the name of the operating system under which BASIC is to run. If you intend to run RTOS, skip the rest of this section and go directly

to "RTOS System Generation," later in this chapter.

3. *System save file (BASIC, any disk filename)?*

Pick a filename for the system you are generating. This may be BASIC or any other name. The name entered becomes the root for system filenames:

- *name.SV* (executable core image file)
- *name.OL* (overlay file)
- *name.CL* (command line file)
- *name.RB* (relocatable binary file)
- *name.DL* (copy of system generation dialog for system being configured)
- *name.MP* (load map file)
- *name.SW* (swapping file)
- *name.AF* (accounting file)

4. *Accounting (YES,NO)?*

Answer yes if you want the system to create a file to record accounting information. *Name.AF* becomes the name of the accounting file for the background and *Fname.AF* for the foreground, where *name* is the name of the system save file.

This option provides useful information about system use, including the times users sign on or off, how much CPU time they use, and how many I/O operations they perform. When a user logs on or off, the system displays information about the user's current session and puts this information in a master file for the system manager.

Because the master accounting file continues to grow and occupy more space, it should periodically be printed or dumped onto tape, then deleted to start fresh.

5. *Multi-user system (NO,YES)?*

Answer no if you are tailoring a single-user system. Respond yes for a multiuser system. If you respond no, the BSG program skips all queries in the BSG dialog that refer to a multiuser system. If you respond yes, you should have QTY, ULM or ALM support in your RDOS system. If you have no such RDOS support, multiuser BASIC is generated; this wastes memory space by generating a larger BASIC than you need.

6. *Line configuration:*

This prompt appears only if the answer to question

5 is yes. BASIC supports lines 0-31 (a maximum of 32 lines). Respond with a list of numbers (separated by commas), a range of line numbers, or both. For example, you can specify lines 0, 1, 2, and 4 with either of the following entries:

0,1,2,4
0-2,4

7. *Master console line number (-1,0-31)?*

This question appears only for a multiuser system. Enter -1 to specify the master terminal. This is usually the system terminal the RDOS CLI uses for its communication: \$TTI/\$TTO in the background, \$TTI1/\$TTO1 in the foreground. In certain extraordinary cases, however, you may wish to assign the master terminal to a multiplexor line. To do this, enter a line number within the range specified in question 6. If you choose this option, BASIC assumes that you have no system terminal (e.g., BASIC in the foreground without a foreground terminal) and avoids doing any I/O to the missing hardware, as this would hang the system. Thus, if you do have a system terminal that is not the master terminal specified by this question, the system terminal remains dead while BASIC is running except for CTRL-A, CTRL-C, and CTRL-F processing.

8. *Is master console a CRT (YES,NO)?*

BSG needs to know how to assign default terminal characteristics, including DSP (disable spooling), DNF (disable nulls after form feeds and carriage returns), BSP (backspace), and XON (x-on or x-off characteristics). If you answer yes, DSP, DNF, and BSP will be ON and XON will be off. If you answer no, XON will be ON and DSP, DNF, and BSP will be off.

9. *Central processor option support:*

Be sure to generate for hardware multiply/divide and hardware floating point if you have them. In particular, microNOVA computers always have hardware multiply/divide and ECLIPSE S250 computers always have floating point instructions. However, the BSG program doesn't have a table of machine features. Thus, if you generate one of these options into the software but don't have the necessary hardware, BSG does not complain; but the BASIC interpreter subsequently issues an explicit warning message.

On the other hand, failing to generate software support for the hardware you have causes serious performance degradation. However, using software

floating-point emulation can be a valuable fallback or "fail-soft" position in case of a floating-point processor hardware malfunction.

a. *Dial-up lines (NO,ALM,4060)?*

This question appears only if the answer to question 5 is yes. Answer no if you are not using dial-up lines; select ALM (the ULM-5 fits under this answer) or 4060, as appropriate, if you are using dial-up lines. (RDOS has its own support for the ALM; BASIC provides support for the 4060 QTY.)

4060 dial-up imposes a time limit on each remote logon. Users who fail to begin a logon within 15 seconds will be disconnected and will have to make another telephone call. ALM dial-up users must press the data button on the data set or insert the head set into the acoustic coupler within 15 seconds or be disconnected. This time limit helps prevent wrong numbers from keeping telephone lines busy.

In some cases, however, the user may need more than 15 seconds to log on. This limit cannot be changed for ALM users, but it can be changed for 4060 users.

The external symbol STIME is the address of the time-out limit in *name.SV* for 4060 users. After the BSG program (as well as the RLDR program) is complete, the time limit may be changed by using an octal or symbolic editor, with the load map, or by using the patch utility.

b. *Swapping (YES,NO)?*

This question appears only if you are tailoring a multiuser BASIC system. Answer no if you do not desire swapping, yes if you want swapping. The swapping files, *name.SW* for the background and *Fname.SW* for the foreground, are built in the default directory.

If you are running two BASIC systems, one in the background and one in the foreground, each system is given a different name as described above. This must be done to run your multiuser BASIC systems simultaneously.

If you answer no to swapping, the available user program size is divided evenly among the number of lines generated. If you answer yes, user swapping is always to disk on unmapped

machines, and to extended memory or disk on mapped machines, depending on the amount of extended memory available.

c. *Hardware multiply/divide (NO,YES)?*

This question appears only if you're using a NOVA computer. This feature is standard on ECLIPSE computers. Answer yes if you have the hardware multiply/divide option on your computer.

d. *Hardware floating point (NO,SING,DOUB)?*

If you have hardware floating point in your system, you must specify whether it is to run as a single- or double-precision unit. A double-precision variable gives greater accuracy but uses twice as much memory to store a value as a single-precision variable. For more information on single and double precision, see the *Extended BASIC Reference*.

If you answer no, the program creates a single-precision, software floating point system. Users with hardware floating point typically choose double-precision.

If you wish to use double-precision variables in your BASIC, you must have hardware or firmware floating point on your system and must answer DOUB to this prompt. There are no software routines for double-precision support.

e. *Memory management/protection (NO,YES)?*

DOS users and RDOS users with original NOVA computers are not asked this question. Indicate whether your machine has this optional feature.

10. *User written subroutines (NO,YES)?*

Indicate whether you will include user-written assembly language subroutines in your BASIC system. If you answer yes, then a relocatable binary file (SBRTB.RB) must be supplied. For descriptions of linking, calling, and storing assembly language subroutines, see the *Extended BASIC Reference*.

11. *String arithmetic (NO,YES)?*

Indicate whether you want to include string arithmetic in your system. If you answer yes, the following question appears:

a. *Portion as overlay (YES,NO)?*

If you answer no, all string arithmetic is put

in main memory. If you answer yes, part of string arithmetic is overlaid, but part also remains in main memory. You lose some speed, but you save about 1000 bytes of user program space. You should carefully consider your speed and space needs before you answer this question.

12. *Default features acceptable (YES,NO)?*

(See question 13.)

13. *Default overlays acceptable (YES,NO)?*

This question appears only if you answered yes to question 12. If you answer yes to both questions 12 and 13, the program assumes the default selection for questions 14a through 14w and does not display them at your terminal. If your answer to either question is no, BSG then asks each question.

In order to increase the available user program space, portions of the BASIC interpreter have been placed in an overlay file and are loaded into main memory only as needed. For seldom used features, this saves substantial user space while decreasing performance only slightly, due to the time needed to load an overlay into main memory. However, if you use an overlay often, the load time may have a serious effect on performance. In such cases we recommend you place the overlay in main memory by answering M to the appropriate questions in 14a-14w.

Responding no to a feature (except string arithmetic) saves space only in the overlay file, not in user memory. Consider carefully such features as PRINT USING and SIN, COS, and TAN, which, if heavily used and not made resident, can exact a severe performance penalty.

14. Valid answers to the following are:

O= Include feature as an overlay.

M= Include feature in main memory.

NO= Do not include feature.

a. *Matrix arithmetic (O,M,NO)?*

Includes: Matrix assignment, addition and subtraction, zero matrix (ZER), multiplication (MUL), unit matrix (CON), determinant (DET), identity matrix (IDN), transposition (TRN).

b. *Matrix input and output (O,M,NO)?*

Includes: MAT READ FILE
MAT INPUT [FILE]
MAT PRINT [FILE]
MAT WRITE FILE

- c. *Matrix inverse (O,M,NO)?*
Matrix inversion (INV)
- d. *PRINT USING (O,M,NO)?*
Includes PRINT USING logic.
- e. *BYE (O,M)?*
- f. *CALL (O,M)?*
Question 14f is relevant only if you answered yes to question 10.
- g. *Mark sense card reader (O,M,NO)?*
If you have a card reader and user abbreviated format DGC mark-sense cards, answer O or M. Otherwise answer NO.
- h. *LIST,PUNCH (O,M)?*
- i. *File commands (O,M,NO)?*
- j. *USERS command (O,M)?*
- k. *Operator commands (O,M)?*
Includes: DISABLE, ENABLE, FREE, KILL, MAX, MSG, ALL, FALL, FMSG, LEVEL
- l. *NEW,AUDIT (O,M)?*
- m. *User initialization (O,M)?*
Called each time a user logs on
- n. *DELETE,RENAME,PAGE,RENUMBER,SIZE,INIT,RELEASE (O,M,NO)?*
- o. *Console characteristics (O,M,NO)?*
- p. *DISK and directory commands (O,M,NO)?*
- q. *BASIC edit and help commands (O,M,NO)?*
- r. *Error message text (O,M,NO)?*
- s. *I/O error message text (O,M,NO)?*
- t. *SIN,COS,TAN,ATN,SYS (O,M,NO)?*
- u. *LOG,EXP,^,SQR,RND (O,M,NO)?*
- v. *LEN,POS,STR\$,VAL,ORD,CHR\$ (M,O,NO)?*
- w. *RUN filename,CHAIN,SAVE,LOAD,logon (O,M,NO)?*
(Logon is the auto logon feature.)

- 15. *Above overlays virtual (YES,NO)?*

This question appears only if the answer to question 9e is yes. If you answer yes, the above overlays are made virtual. If you answer no, you receive the usual system with disk overlays.

If you answer yes, the ground the system is run in must contain enough memory — at least 512K bytes — for the overlay file, or “ERROR 2 - Overlay file” results when BASIC is invoked. Any remaining extended memory is automatically used for swapping, as usual. Virtual overlays increase the speed of overlay loading.

- 16. *Overlay statement compiler (NO,YES)?*

If you answer yes, the statement compiler is overlaid with part of the interpreter to give a larger user area. However, this results in less throughput if program development and production are mixed. If you answer no, you get the regular system with the statement compiler in main memory.

- a. *As virtual overlay (YES,NO)?*

This question appears only if the answers to questions 9e and 16 are yes. If you answer yes, the overlaid statement compiler is also made a virtual overlay. This increases speed (because virtual overlays are faster than disk overlays), but the system may not perform as well in mixed development and production systems as it would if you answered no to question 16. You must answer yes to this question if you answered yes to question 15, and no if you answered no to 15.

- 17. *Maximum files to be open at one time (n,l-m)?*

This question determines the total number of open files that may be active at one time. This includes all files opened by all users as well as files used for SAVE, LOAD, LIST, ENTER, FILE, DISK and NEW filename commands. The system supplies values for *n*, the default, and for *m*. You do not need to figure out how many of your channels will be used to maintain BASIC.

The value of *n* is either two times the number of users or 10, whichever is larger. In an unmapped system each channel requires 76 bytes of memory. *N* is the proper response if you do not expect to do much file I/O.

Your response can range from 1 to *m*. The value of *m* normally is ten times the number of users. This is the largest number of files that may be active at once. However, RDOS imposes a limit of 255

channels; this must also be used for tasks such as swapping and I/O to and from users' terminals. For configurations with many terminals, the total number of channels needed could exceed 255. In such cases, *m* is the channels left for file I/O after subtracting from 255 the minimum number of channels needed to run the system. This number obviously varies from system to system, depending on the number of terminals and users.

You must not exceed the number of channels specified in the mapped RDOS system generation program for your ground, or you get an error message when you invoke BASIC.

18. *File sharing (YES,NO)?*

A yes response to this question gives users the capability of accessing up to eight files located outside their home directories, provided they establish links to the appropriate files and use the CHATR or SHARE commands to make the resolution files sharable. A no response restricts users to files in their own directories.

19. *FOR-NEXT nesting limit (8,4-20)?*

Increasing the nesting level, with single precision, uses 6 words of main memory per level per user; with double precision, 10 words per level per user.

20. *GOSUB nesting limit (8,4-20)?*

Increasing the depth of nesting uses one word of main memory per level per user.

21. *User function nesting limit (6,0-10)?*

Increasing the nesting depth, with single precision, uses 3 words of main memory per level per user; with double precision, 5 words per level per user. If you specify 0, then you may not use any user functions in your programs.

22. *Maximum user core size (in bytes)?*

This question appears only if your answers to both 5 and 9b are yes. This is a useful optimization feature. We recommend you use it, not only to decrease the amount of swapping that might occur, but also to allow swapping to overlap with useful processes.

The default response or a specification of 0 means no artificial limit is set: each user can get up to the total amount of physical memory available. If a limit is specified, each user can get up to that limit or all available memory, whichever is less.

We recommend that at least one-fourth of all active

users be able to fit in the available real or extended swapping area at one time. If at least two users fit in memory, then swapping of one user overlaps with execution of others. In some environments you can relax this restriction, but do it carefully to ensure good performance.

The recommended procedure for making use of this feature is as follows:

1. Generate a test BASIC system without core size limit. Note the result of your SIZE command.
2. Make sure the core area available is large enough to fit one-fourth of your active users. The more active users that can fit in real or extended memory, the better the performance.
3. If users report that they don't have enough room, you can increase the size of the active area. Do this cautiously, however, to avoid serious drops in response time.

23. *Prompt string (*-space-null, any 10 character string)?*

Select the default prompt (*) or choose your own prompt string of up to 10 characters.

Each prompt string must end with one of the following valid delimiters: null, carriage return, or form feed. The BSG program includes this delimiter as the last character in the prompt string and as one in the 10-character limit.

24. *Time of day with prompt (NO,YES)?*

Answer yes if you want the time of day output with each prompt.

After you have answered the queries of the Extended BASIC system generation dialog, the BSG program creates three files, where *name* is your response to question 3:

- *Name.RB* is the BASIC system configuration module.
- *Name.CL* contains a relocatable load command line of the modules created during system generation. This file makes it unnecessary to invoke BSG to reload the .SV file.
- *Name.DL* is a copy of the BASIC system generation dialog and your responses used by BSG to create your system.

BSG automatically executes *name.CL* after the last system generation question. It loads only those modules that are specified in *name.CL*. Execution of *name.CL* creates *name.SV* (the BASIC system save file), *name.OL*

(the system overlay file), and *name.MP* (the system load map file).

It is not necessary to invoke BSG to reload a duplicate of the .SV file; instead, delete the old load map (*name.MP*) and execute *name.CL* indirectly (e.g., *@name.CL@*). This is useful if you wish to create a BASIC system using new build files (e.g., a new update of BASIC), or change the SBRTB.RB file to include assembly language subroutines.

Creating BASIC Disk Directories and Files

Before invoking your generated BASIC system under RDOS or DOS, you must create some BASIC library and user directories and files. These are described below, after a brief general description of RDOS disk partitions and directories.

RDOS permits the parceling of disk file space among several users, on both a fixed and variable basis (see Figure 1-1.) *Secondary partitions* are fixed parcels of contiguous file space and are mutually exclusive subsets of total disk file space, or *primary partition*.

You can allocate file space within any partition on a variable basis. That is, you may allocate mutually exclusive portions of a partition, and these portions may shrink or grow within the limits of total file space available in the parent partition. These variable parcels of file space are called *subdirectories*.

Generally you should allocate fixed parcels of disk space (secondary partitions) to users of a timesharing system while keeping BASIC.DR, the system library directory, in a subdirectory. This allows the library to expand or contract as demand on the system requires, thereby increasing efficiency. The BASIC system files BASIC.PR and BASIC.OL should reside in and be executed from the primary partition.

The commands for creating, changing, initializing, and releasing subdirectories and partitions — CDIR, CPART, DIR, INIT, and RELEASE — are available under BASIC, as well as under the RDOS or DOS CLI. These commands are *privileged*. You may execute them only from the master terminal; they are not available to users. Appendix B summarizes these and other Extended BASIC commands. For more complete information about disk files, directories, and partitions, see the *RDOS Reference Manual* and the *Extended BASIC Reference*.

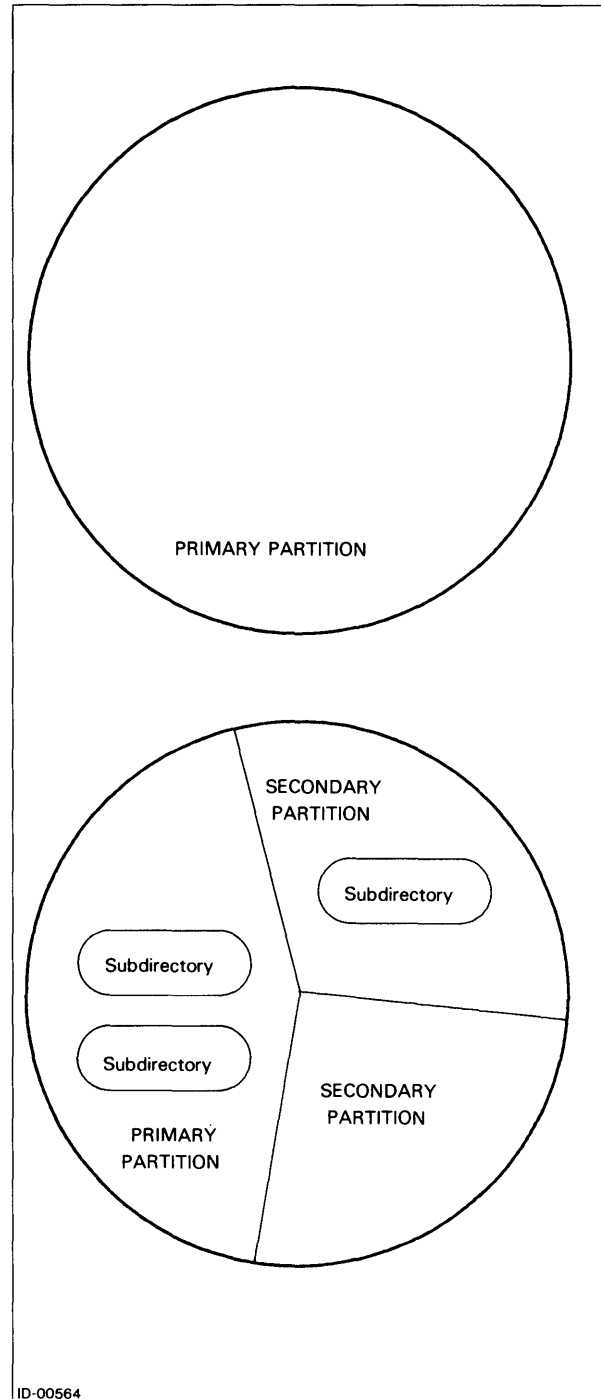


Figure 1-1. Disk File Space Before and After Partitioning

Library Directory: BASIC.DR

The BASIC system library directory, BASIC.DR, must always exist as either a secondary partition or a subdirectory. If it is not initialized, BASIC automatically initializes it, provided that it is a subdirectory and that the BASIC system is contained in its parent partition.

Create the library directory with a BASIC CDIR or CPART command. This directory can include frequently used BASIC utility programs and user-donated programs. User logon programs are saved in BASIC.DR.

The system manager or any user in BASIC.DR may save or list programs into BASIC.DR. You can then use LOAD, ENTER, or RUN on these programs from any user's terminal and, in that respect, all users can share the programs.

User Directories

Assign each user a *user directory*. This is a secondary partition or subdirectory that provides storage space for the user's programs and data files. Create each user directory in the system with a CDIR or CPART command. We recommend you use secondary partitions to control the maximum amount of disk space each user can use.

BASIC.ID File

On multiuser Extended BASIC systems operating under DOS or RDOS, a disk file must be present to contain the account identification of all system users. You can create and modify the BASIC.ID file by using a BASIC program or by using one of the utility text editors provided with RDOS. The file, however, must exist before BASIC is invoked.

The BASIC.ID file should not reside in the BASIC.DR directory or in any directory that users can access. If it does, users can access the file at will and thereby thwart system security.

Each account identification entry in the BASIC.ID file must fit on a single line and be terminated by a carriage return. The format for each line and argument definitions are as follows:

`iden/ [password]//[dev:] dir][filename]`

iden The user account identification (ID). The range is 4-20 alphanumeric characters (0-9 and A-Z only). Each identification entry in the file must be unique.

password An optional four-character alphanumeric password.

dev Required if the directory is not in the same partition as the master device.

dir The optional name of the secondary partition or subdirectory assigned to the account as the user's directory.

It is inadvisable to assign more than one user to a directory since doing so defeats the BASIC file security system.

If the directory name is omitted, the system creates the user's files in the default directory, with the first four characters of the account name prefixed to each filename created. Because RDOS filenames are limited to 10 characters plus a suffix, accounts without an assigned directory are limited to six-character names plus a suffix.

filename The name of the optional logon program saved in the library directory, BASIC.DR. The system automatically executes any logon program included in the user's account identification after a successful logon.

The contents of a BASIC.ID file might appear as follows:

```
MARY/SUB1/PETE
1111/JOES/HALD
DALE//JOHN/HELLO.CI
0002/PROG/USER2
0003//USER3/HELLO
```

Logon Programs

There are several reasons to include an optional logon program with the user identification in the BASIC.ID file:

- A user wants a specific program to execute at each logon.
- The system manager wants a general program to execute each time a user logs on.
- A combination is desired, where the system manager's general program runs, followed by the user's specific program.

Save the logon program in the library directory, BASIC.DR, and include the program name with the user's identification entry in file BASIC.ID. As soon as the user has completed a successful logon, the system immediately executes the logon program.

You must be sure to give the system logon file a unique filename. BASIC first checks the user profile for a logon file, then checks BASIC.DR. If the logon program filename is not unique, the system will never reach BASIC.DR.

To prevent interruption of a logon program, BASIC always disables the recognition of the ESC key just before program execution begins. Therefore, if you desire keyboard interrupt privileges, include a CHAR statement enabling escapes in your logon program or at the end of the BASIC.DR logon program.

```
0005 REM - COMMON LOGON PROGRAM
0010 REM - (HELLO.CI)
0015 IF SYS(11) < 12 THEN GOTO 0040
0020 IF SYS(11) < 18 THEN GOTO 50
0025 GOTO 0060
0040 PRINT "GOOD MORNING"
0045 GOTO 0065
0050 PRINT "GOOD AFTERNOON"
0055 GOTO 0065
0060 PRINT "GOOD EVENING"
0065 IF CPU(0) THEN GOTO 0085
0070 PRINT "SYSTEM GOING DOWN FOR";
0072 PRINT "MAINTENANCE"
0075 PRINT "COME BACK AT 14:30"
0080 BYE
0085 END
```

BASIC.AF File

If you answer yes to BSG question 4 (ACCOUNTING?), the system creates random file *name.AF* for the background and/or *Fname.AF* for the foreground. (*Name* is the system save file you specified in BSG question 3.) If this file already exists, it is retained and all new information is appended. The accounting file grows as BASIC logons and logoffs are executed. Process and delete this file periodically to keep from using too much disk space.

The system writes each entry to the .AF file as a one-line string of ASCII characters. Four lines are written for each user, one at logon, three at logoff. Any of the lines for various system users may be interleaved with lines for other system users. However, each information line has a precise meaning and uniquely identifies a user. Therefore, it is easy to recover the accounting information stored by writing a BASIC program that reads this file. Since each line in the file is ASCII text, it is easy for a BASIC program to process the data. Because there is a space between each field, string functions can readily access the subentries.

The format for lines and definition of items in the .AF file are as follows:

```
iden mm/dd/yy hh:mm SIGN ON,nn
iden mm/dd/yy hh:mm SIGN OFF,nn
iden mm/dd/yy hh:mm CPU USED,t
iden mm/dd/yy hh:mm I/O USED,s,s1
```

iden The user ID of 4 alphanumeric characters. OPER is the system terminal.

mm/dd/yy The date: month/day/year (e.g., 08/15/83)

hh:mm The time in 24-hour format: hour:minute (e.g., 14:25)

nn The port number (e.g., 03; -1 for system terminal)

t The integer number of seconds of CPU time used since logon for the port

s The total number of BASIC file I/O statements (e.g., OPEN FILE, READ FILE, WRITE FILE)

s1 The total number of BASIC I/O statements executed (e.g., ENTER, SAVE DISK, WHATS)

An example of a segment of an .AF file follows:

```
OPER 09/23/83 12:20 SIGN ON,-1
0000 09/23/83 12:21 SIGN ON,01
0001 09/23/83 12:21 SIGN ON,00
0000 09/23/83 12:22 SIGN OFF,01
0001 09/23/83 12:22 SIGN OFF,00
0000 09/23/83 12:23 CPU USED,15
0001 09/23/83 12:23 CPU USED,4
0000 09/23/83 12:23 I/O USED,1,10
0001 09/23/83 12:23 I/O USED,0,0
0000 09/23/83 12:23 SIGN ON,01
0001 09/23/83 12:24 SIGN ON,00
0001 09/23/83 12:27 SIGN OFF,00
0000 09/23/83 12:27 SIGN OFF,01
0001 09/23/83 12:27 CPU USED,31
0001 09/23/83 12:27 I/O USED,1,12
0000 09/23/83 12:27 CPU USED,45
0000 09/23/83 12:27 I/O USED 3,12
OPER 09/23/83 12:27 SIGN OFF,-1
OPER 09/23/83 12:27 CPU USED,0
OPER 09/23/83 12:27 I/O USED,0,14
```

Installing Extended BASIC on RTOS

If you are licensed for RDOS or DOS Extended BASIC, you can also generate a version of BASIC that runs without a direct access device. The Real-Time Operating System (RTOS) is a memory-resident system that does not require a disk. Also, Extended BASIC under RTOS does not support disk memory files. You need at least a NOVA or ECLIPSE central processor unit, a program read device such as a magnetic tape unit, and an input device such as a display terminal.

Before you generate a BASIC system under RTOS, you must first generate an RTOS trigger and necessary system libraries. See the *RTOS Reference Manual*.

RTOS System Generation

Extended BASIC runs only with an unmapped version of RTOS. It does, however, run properly on any mapped or unmapped NOVA or ECLIPSE computer, if you switch off the map during RTOS system generation (RTOSGEN).

Single users under RTOS do not need a real-time clock, but then they cannot use timed commands. Multiuser systems require a real-time clock.

Running Extended BASIC under RTOS requires at least:

- 1 task for the master console reader
- 1 task only for the QTY monitor (multiuser systems only)
- 1 task for each QTY line (compiler/interpreter)
- 1 task for the system operator (compiler/interpreter)

You also need at least:

- 1 channel for the QTY monitor (multiuser systems only)
- 2 channels for the master console (\$TTI,\$TTO)
- 1 channel for each file that you expect to be opened by user programs at the same time
- 1 channel for each BASIC command that causes a file to be opened (SAVE, ENTER, CHAIN, etc.) and that you expect to be executed by users at the same time

RTOSGEN

The following dialog illustrates an Extended BASIC RTOSGEN session:

RTOSGEN

RTOS REV 6.00 SYSTEM GENERATION

MAPPED SYSTEM (0=NO,1=YES)? 0

If you intend to use Extended BASIC on your RTOS system, answer 0. The system ignores the map in the computer and allows BASIC to run properly.

CORE STORAGE (IN K WORDS)? 32

Always answer 32. Extended BASIC under RTOS requires 32K words of storage.

RTC FREQ (0=NONE, 1=10 HZ, 2=100HZ, 3=1000HZ, 4=LINE, 6=INTERNAL)? 1

You must pick a time if you have a multiuser system. 1 or 2 are the most often used frequencies.

TASKS (1-255)? 8

CHANNELS (1-63)? 12

On a typical four-terminal system you need at least eight tasks to run Extended BASIC: four to keep the system up and one QTY line for each terminal (see "RTOS System Generation" above).

You also need at least one channel for each QTY console. Other channels are used to increase the number of files users are allowed to open at the same time (see "RTOS System Generation" above).

This is for a typical four-terminal system. If you use more or less than four terminals, or have special file access needs, you may easily manipulate RTOS to fit your needs.

RESPOND WITH NUMBER OF UNITS

DSK (0-1)? 0

Under RTOS Extended BASIC, there is no disk memory file support.

DKP (0-4) [FOR MODEL 4234, (0-8)]? 0

RTOS BASIC does not support fixed-head disks either.

MTA (0-8)? 1

BUFFERED I/O (1=YES,0=NO)? 1

To generate RTOS you need a program load device such as a magnetic tape. You also need buffers to store I/O.

CAS (0-8)? 0

This question refers to tape cassettes. It is doubtful you would use them with an RTOS system equipped with magnetic tape.

PTR (0-2)? 1

Depending on your needs you may wish to include a high-speed paper tape reader among your peripherals.

PTP (0-2)? 1

If you have a paper tape reader, you should also have a paper tape punch.

LPT (0-2)? 1

COLUMN SIZE (80,132)? 80

Every system should have a line printer to provide users with a hard copy of their files. Column size, which is the length of the line printer line, is up to you. However, 80 conforms to standard 8.5 by 11 inch paper.

CDR (0-2)? 1

Answer with the number of card readers you will hook up to the system.

PLT (0-2)? 1

Answer with the number of incremental plotters you will hook up to the system.

QTY (0-64)? 64

Indicate the number of QTY lines you have.

TTYS (0-3)? 2

You may want to hook terminals up to your system.

MCA (0-15)? 4

Answer with the number of multiprocessor communications adapters you will hook up to the system.

RESPOND WITH 0 FOR NO, 1 FOR YES

AUTO RESTART? 0

You do not need auto restart.

HIGH PRIORITY INTERRUPTS? 0

RTOS is a memory-system. It is unlikely that you will need high-priority interrupts. Everything should work very quickly.

USER SUPPLIED DRIVERS? 0

Not advised for RTOS.

*COMPUTER : NOVA (0), NOVA3 (1), ECLIPSE (2)
OR MICRONOVA (3)? 0*

SUMMARY OF RTOS SYSGEN

CODE DCT NAME NAME

<i>10</i>	<i>TTIDCC</i>	<i>\$TTI</i>
<i>11</i>	<i>TTODC</i>	<i>\$TTO</i>
<i>12</i>	<i>PTRDC</i>	<i>\$PTR</i>
<i>13</i>	<i>PTPDC</i>	<i>\$PTP</i>
<i>14</i>	<i>RTCDC</i>	
<i>17</i>	<i>LPTDC</i>	<i>\$LPT</i>
<i>22</i>	<i>MTADC</i>	
<i>30</i>	<i>QTYDC</i>	
<i>50</i>	<i>TTIID</i>	<i>\$TTII</i>
<i>51</i>	<i>TTOID</i>	<i>\$TTOI</i>

SYSGEN OKAY? 1

OUTPUT FILENAME?

BASIC System Generation Dialog Under RTOS

Once you have built your RTOS system, you are ready to generate BASIC. Following is the system generation dialog needed to build multiuser BASIC on a NOVA system with 4060/4026 remote station support:

BSG

Extended BASIC Revision xx.xx mm/dd/yy System Generation

Valid answers are in parentheses. Default values are given first and may be selected by a null response (carriage return). Defaults are followed by a list of alternatives, one of which may be selected by entering it as it appears in the prompt and pressing the carriage return.

1. *Object computer type (NOVA,ECLIPSE,ORIGINAL)?*

Respond with the type of computer the BASIC system is to run on. ORIGINAL refers to the original NOVA processor. This dialog is based on use with a NOVA computer.

2. *Object operating system (RDOS,RTOS,DOS)? RTOS*

3. *System save file (BASIC, any disk file name)?*

4. *Multi-user system (NO,YES)?*

If you answer no, all the queries in the BSG dialog that relate to multiuser systems are skipped. If you answer yes, you must have already tailored your RTOS system during RTOS system generation for multiusers.

5. *Line configuration: 0-31*

This question appears only for multiuser systems. Respond with a list of numbers, a range of line numbers, or both. BASIC supports lines 0-31 (a maximum of 32 lines). List the line numbers, separated by commas, or give a range of line numbers. For example, you can specify lines 0, 1, 2, and 4 with either of the following entries:

0,1,2,4

0-2,4

6. *Master console line number (-1,0-31)?*

This question appears only for a multiuser system. Enter -1 to specify the master terminal. This is usually the terminal the RDOS CLI uses for its communication, but in certain extraordinary cases you may wish to assign it to a multiplexor line. This line number must be within the range you specified in question 5. If you choose this option, you cannot use the CLI console (\$TTI/\$TTO in background; \$TTII/\$TTOI in foreground) as a BASIC user terminal.

7. *Is master console a CRT (YES,NO)?*

BSG asks how to assign default terminal characteristics (including DSP, DNF, BSP and XON) to the master console. For more information see question 8 in the dialog for generating RDOS and DOS.

8. *Central processor option support:*

a. *Dial-up lines (NO,ALM,4060)?*

This question appears only for multiuser systems. Select no if you are not using dial-up lines; select ALM (the ULM-5 fits under this answer) or 4060, as appropriate, if you are using dial-up lines. RTOS supplies support for the ALM; BASIC supplies support for the 4060 QTY.

4060 dial-up imposes a time limit on each remote logon. Users who fail to begin logon within 15 seconds are disconnected and have to make another telephone call. ALM dial-up line users must press the data button on the data set or insert the head set into the acoustic coupler within 15 seconds or be disconnected. This time limit helps prevent wrong numbers from keeping telephone lines busy.

In some cases, however, the user may need more than 15 seconds to log on. The external symbol STIME is the address of the time-out limit in file *name.SV* for 4060 users. After the BSG program (and hence the RLDR program) is complete, the limit may be changed by using an octal or symbolic editor in conjunction with the load map or by using the patch utility. The limit cannot be changed for ALM users.

b. *Hardware multiply/divide (NO,YES)?*

This question appears only if you are using a NOVA computer. This feature is standard on ECLIPSE computers. Tell whether or not your computer is equipped with the hardware multiply/divide option.

c. *Hardware floating point (NO,SING,DOUB)?*

Again, this depends on your hardware configuration. If you have hardware floating point in your system, you must specify whether it is to run as a single- or double-precision unit.

If you answer no, BASIC builds a single-precision software-driven system.

9. *User written subroutines (NO,YES)?*

Indicate whether you will include user-written assembly language subroutines in your BASIC system. If you answer yes, then a relocatable binary file (SBRTB.RB) is created. For descriptions of linking, calling, and storing assembly language subroutines, see the *Extended BASIC Reference*.

10. *String arithmetic (NO,YES)?*

Answer yes if you want to include string arithmetic in your RTOS system. String arithmetic under RTOS is core resident only. Carefully consider your user's space requirements before you answer this question.

11. *Default features acceptable (YES,NO)?*

If you answer yes, the system includes the default selection of questions 12a through 12s without printing the questions at your terminal. If your answer is no, then the BSG program asks you to respond to each question in 12.

12. Valid answers to the following are:

M=Include feature in main storage
NO=do not include feature

a. *Matrix arithmetic (NO,M)?*

Includes: Matrix assignment, addition and subtraction zero matrix (ZER), multiplication (MUL), unit matrix (CON), determinant (DET), identity matrix (IDN), transposition (TRN)

b. *Matrix input and output (NO,M)?*

Includes: MAT READ FILE
MAT INPUT [FILE]
MAT PRINT [FILE]
MAT WRITE FILE

c. *Matrix inverse (NO,M)?*

Matrix inversion (INV)

d. *PRINT USING (M,NO)?*

Includes PRINT USING logic.

e. *BYE (M,NO)?*

f. *Mark sense card reader (NO,M)?*

If you have a card reader and user abbreviated format DGC mark-sense cards, answer M. Otherwise answer NO.

g. *LIST,PUNCH (M,NO)?*

h. *USERS command (M,NO)?*

i. *Operator commands (M,NO)?*

Includes: DISABLE, ENABLE, FREE, KILL, MAX, MSG, ALL, FALL, FMSG, LEVEL

- j. *DELETE,RENAME,PAGE,RENUMBER,SIZE,INIT,RELEASE (M,NO)?*
- k. *Console characteristics (M,NO)?*
- l. *BASIC EDIT and HELP commands (NO,M)?*
- m. *Error message text (M,NO)?*
- n. *I/O error message text (M,NO)?*
- o. *SIN,COS,TAN,ATN (M,NO)?*
- p. *LOG,EXP,|,SQR,RND (M,NO)?*
- q. *LEN,POS,STR\$,VAL,ORD,CHR\$(M,NO)?*
- r. *RUN filename, CHAIN, SAVE,LOAD,logon (M,NO)?*

(Logon is the auto logon feature.)

13. *FOR-NEXT nesting limit (8,4-20)?*

Increasing the nesting limit, with single precision, uses six words of main memory per level per user; with double precision, ten words per level per user.

14. *GOSUB nesting limit (8,4-20)?*

Increasing the depth of nesting consumes main memory by one word per level per user.

15. *User function nesting limit (6,0-10)?*

Increasing nesting depth, with single precision, uses three words of main memory per level per user; with double precision, five words per level per user. However, if you specify 0, you may not use user functions in your program.

16. *Prompt string (*-space-null, any 10 character string)?*

Select the default prompt (*) or choose your own prompt string of up to 10 characters. Every prompt string must end with one of the following valid delimiters: null, carriage return, or form feed. This delimiter is included as the last character in the prompt string and counts as one of the 10 characters.

17. *Time of day with prompt (NO,YES)?*

Answer yes if you want the time of day output with each prompt.

BSG creates a relocatable loader command file (in this case called RTOSBASIC.CL) that contains the following commands:

```
RLDR/A/P/E/C/N RTOSBASIC.MP/L RTOSBASIC
QTYM QTYR MDHW PRUY BYE LIST1 OPER1
OPER INIT2 INIT1 MISC1 CHAR ERROR
ERRIO FUNC2Y FNC1Y FUNC3Y EXEC
BASICA.LB BASICB.SLB BASICY.LB TIME2
RTOS NRTOS1.LB NRTOS2.LB INIT
```

This file is sufficient to build an executable file. However, if you need I/O capabilities such as magnetic tape and cassettes, you must modify the RLDR command file to include the appropriate RTOS driver libraries:

```
RLDR/A/P/E/C/N RTOSBASIC.MP/L RTOSBASIC
QTYM QTYR MDHW PRUY BYE LIST1 OPER1
OPER INIT2 INIT1 MISC1 CHAR ERROR
ERRIO FUNC2Y FUNC1Y FUNC3Y EXEC
BASICA.LB BASICB.LB BASICY.LB TIME2 RTOS
MTADR.LB CASDR.LB NRTOS1.LB
NRTOS2.LB INIT
```

The libraries appear immediately after the keyword RTOS in the command file because RTOS does not function properly if they appear in any other order.

The command file may be executed by typing:

```
@RTOSBASIC.CL@
```

Once an executable memory image file has been built and all patches applied, you can execute the file using the RDOS BOOT command. (See the latest RDOS Extended BASIC update notice for patching instructions.) An RTOS BASIC session might look like the following:

```
R
BOOT RTOSBASIC
MASTER DEVICE RELEASED
RTOS REV 6.00
DATE (M/D/Y) ? 11 2 83
TIME (H:M:S) ? 14 33
EXTENDED BASIC REVISION 5.20 08/01/78
OPER 11/02/77 14:33 SIGN ON,-1
* INIT "MTO"
* NEW "MTO:16"
* LIST
0010 FOR I=1 TO 10
```


BASIC Commands, Statements, and Functions on RTOS

The following Extended BASIC commands, statements, and functions work the same under both RDOS and RTOS.

ABS	CALL	ATN	BYE
CALL	CHAR	CHR\$	CLOSE
CON	COS	DATA	DEF
DELAY	DET	DIM	END
ERASE	EOF	EXP	FALL
FMSG	FN	FOR	FREE
GOSUB	GOTO	OF	INPUT
INT	KILL	LEN	LET
LEVEL	LOG	LREAD	LWRITE
MAT	MAX	MSG	NEXT
ON	ORD	PAGE	POS
PRINT	PUNCH	READ	RANDOMIZE
SIZE	SQR	STOP	STR\$
SYS	TAB	TAN	TIME
TINPUT	USERS	VAL	WRITE

Extended BASIC editing commands (.P, .C, etc.) also work in RTOS. For more information on these commands, see the *Extended BASIC Reference*.

The following Extended BASIC commands, statements, and functions work only as reserved (nondisk) files in RTOS:

OPEN	CHAIN	ENTER	LIST
LOAD	NEW	RUN	SAVE
INIT	RELEASE	AUDIT	

The following Extended BASIC commands, statements, and functions do not work under RTOS:

CHATR	DELETE	DISK	FILE
GDIR	GPOS	LIBRARY	RENAME
RESET	SPOS	WHATS	CDIR
CPART	DIR	HELP	

Other functions that require a direct access storage device (such as swapping, accounting, and user directories) are not supported under RTOS.

Any response of four or more characters to the sign-on prompt of ACCOUNT NAME is accepted. The first four characters may thereafter be used as arguments to the MSG, FMSG, FREE and KILL commands and can be displayed with the user's command.

CTRL-A, CTRL-C, and CTRL-F have no meaning in RTOS since it has no command line interpreter and does not require a task. The BASIC commands ENABLE and DISABLE do not work.

RTOS is not compatible with either DOS or RDOS. You may have trouble running Extended BASIC, especially on a multiuser system, for the following reasons:

- The QTY monitor function, QTY:64, does not recognize an escape as an interrupt key. Thus, the only way to interrupt a looping program is with the operator command FREE. However, an ESC typed on the master terminal is recognized.
- The QTY driver does not support CRT rubout, TAB character expansion, character buffering, character echo, etc. BASIC commands such as CHAR and INPUT VAR do not work properly.
- There is no ALM or ULM support.
- Carriage return and form feed keys are not always echoed.
- The QTY rubout character is a backslash.
- Users can reference all eight magnetic tapes and all eight cassettes even though only one was asked for in RTOSGEN.
- Paper tapes cannot be read from terminals attached to a QTY.

Installing Extended BASIC on AOS and AOS/VS

Extended BASIC for AOS and AOS/VS systems requires neither a special operating system nor a BASIC system generation. The release tape or diskette contains both a double-precision and a single-precision version of Extended BASIC that have been built with a default overlay structure (described below). If this structure is adequate, you need only load the release media onto your system. However, the release tape or diskette also contains load files and macros that allow you to relink the Extended BASIC modules to produce a system tailored to specific needs.

Double-precision BASIC works a calculation out to more significant digits than single-precision BASIC; it takes up more, but not significantly more, memory. Typically users choose double-precision BASIC.

Loading BASIC Modules

Following are the instructions for loading Extended BASIC modules from the release tape or diskettes onto your system. Note that *x* stands for the number of the tape or diskette drive.

1. Load the BASICHELP directory into the root (:) directory. To load from tape, the commands are:

```
DIR :
LOAD/V/R @MTAx:0 BASICHELP:+
```

To load from diskette, insert the diskette containing the BASICHELP directory and issue these commands:

```
DIR :
LOAD/V/R @DPDx BASICHELP:+
```

2. Load the released files into the desired directory. The :UTIL directory is recommended for the program (.PR) and overlay (.OL) files. The commands given below assume that all files are loaded into :UTIL. To load from tape, the commands are:

```
DIR :UTIL
LOAD/V/R @MTAx:0 +\BASICHELP
```

The \BASICHELP template prevents the duplicate loading of the BASICHELP files.

To load from diskettes, insert the diskette containing BASIC, BASICSP, and RLS and issue the following commands. Then insert the diskette containing the BASIC binaries and issue the commands again:

```
DIR :UTIL
LOAD/V/R @DPDx +
```

3. Merge the Extended BASIC error message file (ABERMES.SR) with the system error message file (:ERMES). Use the macro provided with your operating system: NEWERMES.CLI for AOS and LINK_ERMES.CLI for AOS/VS. The steps are:
 - a. Edit the link line in the system macro (NEWERMES.CLI or LINK_ERMES.CLI) so that it contains the AB prefix.
 - b. Execute the macro (i.e., type NEWERMES if you have an AOS system or LINK_ERMES if you have an AOS/VS system).
 - c. Move the updated system error message file to the root directory and delete the old one. The command is:

```
MOVE/V/DELETE :ERMES
```

Make sure that the searchlists of all users contain the locations of the library, system, and overlay files.

Both the double-precision and single-precision versions of Extended BASIC are built with the following modules defaulted as overlays:

CALL	User assembly language interface
CHAR	The CHAR command
DIR	FILE, LIBRARY, ACL, DISK, SEARCHLIST, and DIR commands
ENTER	The BASIC statement interpreter
EXEC	LOAD, CHAIN, SAVE, and RUN filename
FUNC1(V,W)	LOG, EXP, I, SQR and RND for double- (V) or single-precision (W) BASIC
FUNC2	SIN, COS, TAN and ATN
INIT1	User initialization, NEW and AUDIT commands
LIST	LIST command
MATAR(V,W)	Matrix arithmetic for double- (V) or single-precision (W)
MATIO(V,W)	Matrix I/O for double- (V) or single-precision (W)
MISC1	PAGE, TAB, SIZE, RENUMBER, HELP, and EDIT commands
OPER	CLI, MSG, and WHO
PRU	PRINT USING
STAR	String arithmetic

In most cases the released configuration is adequate. However, to call user-written assembly language subroutines or to change the default overlay configuration, you must edit the BASIC load line and relink BASIC. These procedures are explained below.

Calling Assembly Language Subroutines

To be able to call user-written assembly language subroutines, first put the routines in a source file called SBRTB.SR. Then use the macroassembler, MASM, to produce the object module SBRTB.OB. More information about MASM is available in the *Macroassembler Reference Manual* for your operating system (093-000192 for AOS; 093-000242 for AOS/VS). The *Extended BASIC Reference* explains the interface between BASIC and assembly language subroutines.

Next, edit the BASIC load line file — LOAD.CL for double- and LOADSP.CL for single-precision BASIC — to include SBRTB. For example, the released double-precision load line file, LOAD.CL, contains the following:

```
!*CALL!CHAR!DIR!ENTER!EXEC!FUNC1V!FUNC2
INIT1!LIST!MATARV!MATIOV!MISC1!OPER!PRU!STAR*!
CNVRTV EDML ENTRY EVEXV FILES FPMISV
FUNCV FUNC3 INIT MISC PI RUN STATEV SVRS
SYNTAX USER ZREL V
```

After you have edited LOAD.CL to include SBRTB, it should contain the following:

```
!*CALL!CHAR!DIR!ENTER!EXEC!FUNC1V!FUNC2
INIT1!LIST!MATARV!MATIOV!MISC1!OPER!PRU!STAR*!
SBRTB CNVRTV EDML ENTRY EVEXV FILES
FPMISV FUNCV FUNC3 INIT MISC PI RUN
STATEV SVRS SYNTAX USER ZREL V
```

Now use the supplied macros to relink BASIC: enter MAKE_BASIC to build a double-precision BASIC.PR file and MAKE_BASICSP to build a single-precision BASIC.PR file.

Changing the Overlay Configuration

The filenames contained in LOAD.CL and LOADSP.CL are the names of the modules that are bound together to make BASIC.PR and BASICSP.PR. The modules enclosed between !* and *!, separated by !, are placed in the BASIC overlay file, while the modules separated by blanks are placed in main memory. To change this configuration, edit the load line file. For example, to generate a single-precision BASIC with the overlay routines CALL and PRU in main memory, edit the LOADSP.CL file, which initially contains the following:

```
!*CALL!CHAR!DIR!ENTER!EXEC!FUNC1W!FUNC2!
INIT1!LIST!MATARW!MATIOW!MISC1!OPER!PRU!
STAR*!CNVRTW EDML ENTRY EVEXW FILES
FPMISW FUNCW FUNC3 INIT MISC PI STATEW
SVRS SYNTAX USER ZREL W
```

After editing, LOADSP.CL should contain the following:

```
!*CHAR!DIR!ENTER!EXEC!FUNC1W!FUNC2!
INIT1!LIST!MATARW!MATIOW!MISC1!OPER!STAR*!
CALL PRU CNVRTW EDML ENTRY EVEXW FILES
FPMISW FUNCW FUNC3 INIT MISC PI RUN
STATEW SVRS SYNTAX USER ZREL W
```

Then enter MAKE_BASICSP; this macro links a BASIC system with CALL and PRU in the BASICSP.PR file, rather than in the BASICSP.OL file.

Managing the User Environment

AOS and AOS/VS provide the PREDITOR program, which allows the system manager to assign a name and password to each authorized system user and to specify the privileges of each. Each user then must enter the name and password in order to log on, and is restricted to the privileges assigned via PREDITOR while using the system.

Extended BASIC requires that each user have the IPC privilege. Other privileges and operating characteristics are up to the manager. See *Managing AOS* or *Managing AOS/VS* for detailed information about PREDITOR and security options.

Invoking Extended BASIC

The command for invoking Extended BASIC from the CLI is one of the following:

```
X BASIC      (double-precision)
X BASICSP    (single-precision)
```

AOS and AOS/VS provide the system manager with many options for controlling the working environment. For example:

- Each user may be allowed to execute BASIC directly from the CLI by issuing the command above.
- The command for invoking BASIC may be placed in a macro that sets or alters the environment.
- The user profiles built with PREDITOR may be such that users log directly on to Extended BASIC rather than to the CLI, and log off the system when they exit from BASIC.
- BASIC may be invoked through the batch stream. This option allows the running of BASIC programs without having to enter and exit from BASIC.

See the *Command Line Interpreter (CLI) User's Manual (AOS and AOS/VS)* for information about the choices available for invoking Extended BASIC and controlling the working environment.

Shared Files and Record Locking

Extended BASIC allows more than one user to update a file simultaneously, provided these conditions are met:

- The file already exists before opening it.
- The file must be opened as shared (mode 7). (Note that all pages of a shared file are shared.)
- The element size of the shared file must be a multiple of 4. The Resource Locking Server (RLS) program must be running as a separate process under PID 2.

The command for executing RLS is:

```
PROCESS/DIRECTORY=:UTIL/IPCUSAGE/RESI-  
DENT/CALL=3/NAME=RLS/USER=RLS RLS
```

If file sharing is allowed at the site, it is advisable to put this command in the UP.CLI macro, as doing so ensures that the RLS process is always active. Extended BASIC runs without the RLS process, but a serious problem results should more than one user attempt simultaneously to update the same record in a shared file.

Once executed, RLS runs until explicitly terminated. The command is:

```
TERMINATE/2=IGNORE RLS:RLS
```

This command may be placed in the DOWN.CLI macro or executed from PID 2.

Testing Extended BASIC

The release tape or diskette for Extended BASIC under AOS and AOS/VS also contains the certification programs MAIN.CI (double-precision) and MAINSP.CI (single-precision). These programs execute some simple BASIC statements. We recommend that you run these programs prior to using your BASIC system for development. No errors should result from the programs. If any do, contact the local system engineer. To run the programs, execute your Extended BASIC and issue one of these commands:

```
RUN "MAIN.CI      (double-precision)  
RUN "MAINSP.CI   (single-precision)
```

End of Chapter

Chapter 2

Start-up, Shutdown, and Crash Procedures on RDOS and DOS

This chapter explains how to bring up and shut down BASIC on your RDOS or DOS system. It also explains what to do in case of a system crash or a power failure.

Cold Starting BASIC

A *cold start* is the system start-up procedure to use when the CPU has been halted or turned off in an orderly manner, or after an initial RDOS or DOS system generation and BSG (as described in chapter 1). The restart procedure follows an inadvertent system crash.

System Preparation

1. Turn on the system terminal, the CPU, and the disk unit you're using to boot the system.
2. The disk mounted in the disk unit must have the bootstrap program and a properly generated RDOS or DOS operating system save file (see *How to Load and Generate Your RDOS System* or *How to Generate Your DOS System*).
3. Load the RDOS or DOS system by performing the appropriate disk bootstrap procedure.
4. Proceed using an existing BASIC system save file, or generate a BASIC system as described in chapter 1. Note that more than one BASIC system save file may be available.

Invoking the BASIC System

We recommend that your BASIC system save and overlay files (BASIC.SV and BASIC.OL) reside in the parent partition containing the system library, BASIC.DR, and user directories. This insures that BASIC can find and, if necessary, initialize these subdirectories.

Use the DIR command to access the directory that contains the BASIC system, then type the name of your BASIC system in response to the CLI prompt. For example:

```
R
DIR DPO
R
SMALLBASIC
```

SMALLBASIC.SV is located in the current directory and is the name given to the BASIC system save file.

Once BASIC is running it displays its revision number and the date on which system testing and development of the version was completed:

BASIC REVISION xx.xx mm/dd/yy

At this point BASIC performs a number of initialization tasks before it proceeds. These tasks include initializing the library directory and creating a swapping file, if necessary. If an initialization error occurs, an error message is output to the system terminal; BASIC stops and returns to the CLI. Appendix A lists and describes initialization error messages. Correct any errors before reinvoking BASIC.

The system is now ready for use, provided the BASIC.ID file, the BASIC common library directory (BASIC.DR), and user directories have been created as described in chapter 1.

Stopping the BASIC System

Occasionally it may be necessary to shut down BASIC for maintenance or to generate a system with a different configuration. An orderly shutdown procedure, such as the following, minimizes inconvenience to users:

1. Prevent any new user from logging on:

```
*MAX= 1
```
2. *ALL SYSTEM COMING DOWN IN FIVE MIN PLS

```
*ALL LOGOFF BY THEN. THANKS
```
3. Determine who is still logged on:

```
*USERS
NAMESTATUS...
0000 0002...
OPER 0002...
```

User 0000 is still logged on.
4. Send the user another warning message:

```
*FMSG 0000 PLS LOGOFF
```

5. *BYE
ERROR 64-SYSTEM ACTIVE
The operator cannot kill an active system.
6. *USERS
NAMESTATUS...
0000 0042...
OPER 0042...
7. Force user 0000 off the system:
*KILL 0000
8. Return control to RDOS:
*BYE
System manager's accounting information is displayed:
OPER 11/13/75 17:29 SIGN OFF,-1
OPER 11/13/75 17:29 CPU USED,0
OPER 11/13/75 17:29 I/O USED,0,0

17:29:10

R

Recovery from System Crashes

A system crash is any termination of operation other than release of the master partition (e.g., the system "hangs"). If your system inadvertently fails, you should document the circumstances of the failure before proceeding to reinvoke BASIC.

1. Determine whether the system is halted or looping. If it is halted, record the contents of the PC (program counter) and each of the four accumulators (AC0, AC1, AC2, AC3) and proceed to step 3. If the system is looping, record the range of the loop by repeatedly executing hardware interrupts to randomly sample the loop: press the CONTINUE and STOP switches or, with a virtual console, press the BREAK key and enter P. If your system has an instruction step switch, do not attempt to use it to trace the loop, as you will immediately enter the clock interrupt routine.
2. Attempt to return to RDOS by typing ENABLE at the system terminal (if DISABLE has been previously used) and by entering CTRL-C or CTRL-A. If possible, use CTRL-C and save the resulting core image break file. If this is successful, you have avoided a true system crash and you should proceed to cold start BASIC as described previously.

3. If CTRL-C or CTRL-A does not free the system and return you to the CLI, then take a machine-readable RDOS core dump. If you have a mapped system, an RDOS core dump provides more information than a break file, and should be taken if termination of the other ground is not catastrophic.
4. Document as completely as possible the circumstances of the crash, including loop addresses, break file or core dump, RDOS and BASIC load maps, SYSGEN dialog, and a snapshot of the swapping file, as well as what users were on the system, what peripherals were hooked up or in use, and what programs were active at the time of failure. The documentation should be sent to your local systems engineering office for analysis.

After a system crash, the integrity of the disks in use at that time should be treated as suspect. Under no circumstances should users continue to use a crashed disk without following these steps:

1. Reboot the system.
2. Execute CLEAR/A/V/D for all active directories and disks.
3. Back up each crashed disk on tape or another disk using the CLI DUMP command.
4. Perform a full DKINIT on each crashed disk.
5. Use the CLI LOAD command to restore the files on each crashed disk. (This procedure eliminates disk format and MAP.DR errors.)

The data files are now accessible. However, the data within the file may be corrupted. If a program was reading a file, however, a system crash does not corrupt the integrity of the file.

Updating the Records of a File

If, at the time of a crash, a program was updating the records of a file by first reading and then rewriting them, the integrity of the file is intact after the crash. However, there is a possibility the I/O to the disk was not completed before the crash. After the system is restarted, implement start-up procedures to verify that the last record processed was updated before resuming the updating of the remaining records.

Complications can occur if a single transaction updates more than one record in the same file or in different files. It is possible that, as a result of the crash, some of the records have been updated and others are unchanged. It is also possible that the most recently added records were lost during the crash if the records were in the I/O buffers. However, the preupdate file should be intact.

Auto Restart Procedure

RDOS provides software support for an optional auto restart after a power failure. When power is restored, if the console key is in the LOCK position, you see the following message on the system terminal:

POWER RESTORED

If the console key is in the ON position when input power fails, set the console switches to all zeros (down) and press START when power is restored; with a virtual console, press capital P. This outputs the console message and restores the BASIC system as it does when the key is in the LOCK position.

The following system devices are given power-up restart service:

- Paper tape reader/punches
- Quad multiplexors
- Card readers
- Line printers
- Disks

Character devices may lose one or more characters while power comes up. Each card reader may lose up to 80 columns of information on a single card. Line printers may lose up to a single line of information. Since power-up service for disks includes a complete reread or rewrite of the current disk block, no disk information is lost; however, moving-head disk units require 30 to 40 seconds before disk operations can continue. Devices requiring operator intervention (like line printers, card readers, etc.) must receive such action if power is lost for an extended period of time.

No power-up service is provided for magnetic tape or cassette units.

System Backup Procedures

You should follow a periodic procedure for preserving system and user programs. Such a procedure varies from one installation to the next, depending on hardware configurations and user requirements.

Generally, you should shut down the system during nonprime hours and transfer the contents of each directory in the system to magnetic tape or a disk cartridge set aside specifically for backup. You can make the transfer using the CLI DUMP command. Store tapes or disks used for backup in a secure area away from the immediate location of the computer.

End of Chapter

Chapter 3

Extended BASIC System Operation on RDOS and DOS

In a typical timesharing BASIC system the system manager and the system operator have privileged roles; each must supervise vital parts of the system for its users. This chapter addresses the responsibilities of both operator and manager, including maintenance of, and security measures for, the system on RDOS and DOS.

System Operator Responsibility

The system operator for a timesharing system generally has the responsibility for controlling and maintaining the system's resources on a real-time basis. The operator's duties may include such duties as mounting and dismounting magnetic tapes, distributing line printer output, and updating user identifications to control system access by potential users.

The BASIC system provides one terminal with special capabilities and commands that cannot be executed on any other terminal connected to the system. This privileged terminal is called the master terminal; it is usually the same system used by the CLI. The master terminal offers all the normal features of Extended BASIC and can be used as if it were any other user terminal.

Communicating with Users

You can execute four commands from the system terminal to communicate with active users. A forced message allows you to send a message even if CHAR has been used to disable message reception.

MSG	Message to single user
FMSG	Forced message to single user
ALL	Message to all users
FALL	Forced message to all users

These commands are described in the *Extended BASIC Reference*.

Changing Directories

You can change the current directory with the DIR command. This ability to change directories at the system

terminal allows the system operator to maintain files in any directory in the system, even if the user assigned to the directory is logged on. The system operator can access any file in the directory unless a user has already opened it for I/O.

You can change directories at the system terminal without interrupting normal operation or disturbing other users in a multiuser environment.

System Console Breaks

There are three RDOS system break characters you can input from the system terminal:

- CTRL-A Interrupt background and foreground
- CTRL-C Interrupt and save background and foreground
- CTRL-F Interrupt foreground from background

These break characters can abort the BASIC system, and are described in the *RDOS/DOS Command Line Interpreter Reference Manual*.

When operating BASIC, the CTRL-A and CTRL-C system breaks serve no practical purpose, since BASIC is equipped with an ESC break and a BYE command to return to the CLI. The use of CTRL-A, CTRL-C, or CTRL-F under BASIC can result in the loss of accounting information in the *name.AF* file, and incomplete or erroneous data files.

To prevent the inadvertent use of CTRL-A and CTRL-C use the BASIC commands DISABLE and ENABLE (described in the *Extended BASIC Reference*).

Note that the system break characters listed above have no meaning in BASIC and are ignored if used at a non-master terminal.

Device Independence

In a timesharing environment, there is often more than one of each type of secondary storage and I/O device in the system (e.g., magnetic tape and line printers). And, more often, it is impractical to provide each user with a permanently reserved device. Therefore, the resources of the timesharing system must be shared in some practical

manner that minimizes inconvenience to users and promotes efficient use of system peripherals.

One such technique includes the concept of device independence, where the users' programs are written to defer the actual naming of a peripheral device until program execution time. Typically this is accomplished as shown in the following segment:

```
0100 INPUT "Tape Unit:",T$
0110 LET F$=T$,".5", "<0>"
0120 OPEN FILE (3,2),F$
```

Before running this program, the user sends a message to the system operator, requesting that a specific tape be mounted. The message should include any requirements for removing the write-enable ring. The system operator then acknowledges receipt of the message, locates the user's tape, mounts it on an available magnetic tape unit, initializes the device, and returns a message to the user stating which drive was selected.

The dialog at the system console might be as follows:

```
* FROM BILL: MOUNT TAPE # DO33 WRITE
ENABLE
```

The operator acknowledges the message, mounts and initializes the device, and advises user to proceed:

```
*MSG BILL MSG RCVD.
*INIT "MT5"
*MSG BILL DO33 ON MT5. GO AHEAD
```

Since the operator specified MT5, the user can proceed as follows:

```
*ENTER "PROG"
*RUN
```

Tape Unit:MT5 (query from program)

```
.
.
.
```

After the programming session, the user notifies the system operator to dismount the magnet tape. Dialog at the system console might be:

```
FROM BILL:DONE.REMOVE #DO33 FROM MT5.
*MSG BILL MSG RCVD. HAVE A NICE DAY.
*RELEASE "MT5"
```

System Manager Programs

The following programs are designed to increase system flexibility. They give the system manager the ability to manipulate the BASIC system without having to take BASIC down to make changes. These programs are designed to work only at the master terminal.

The program EDITID.QA allows the system manager to add and delete profiles. It also allows the manager to change passwords, display the BASIC.ID file, change it, and then delete the old BASIC.ID file. The program appears Figure 3-1. Results of the program appear in Figure 3-2.

Your results may differ slightly from those shown since, for instance, your BASIC.ID file probably is different from that in the example. Be aware of these differences as you examine the results of the program.

The program USERS.QA in Figure 3-3 allows you to receive constantly updated accounting information. It shows what users are logged on, what lines they are on, data block usage, CPU used, and other useful information.

Notice that, in the results in Figure 3-4, the time is printed at the top of the example. As the cursor goes down the screen it updates all the information. There is a small time lag in displaying a change in a user's condition, but the difference is not significant.

The program FREEID.QA in Figure 3-5 allows you to free all users and return them to the BASIC.CLI. In the process, it creates a BASIC program, FREEID.TM, which contains all the users, enters it as an overlay, and then frees everybody.

```

* NEW "EDITID.QA"
* LIST
0010 NOESC
0020 DIM A$(40)
0030 DIM D$(40),H$(40),H5$(40),S$(100)
0040 INPUT "WHICH DIRECTORY IS BASIC.ID IN ?",D$
0050 DIR D$
0060 PRINT
0070 PRINT "COMMANDS ARE:"
0080 PRINT "DIS - TO DISPLAY BASIC.ID"
0090 PRINT "ADD - TO ADD AN ENTRY TO BASIC.ID"
0100 PRINT "DEL - TO DELETE AN ENTRY FROM BASIC.ID"
0110 PRINT "CHG - TO CHANGE AN ENTRY IN BASIC.ID"
0120 PRINT "NEW - TO DELTE BASIC.IC"
0130 PRINT "STP - STOP"
0140 PRINT
0150 INPUT "COMMAND ? ", C$
0160 IF LEN(C$)<3 THEN GOTO 0060
0170 IF C$(1,3)="DIS" THEN GOTO 0240
0180 IF C$(1,3)="ADD" THEN GOTO 0310
0190 IF C$(1,3)="DEL" THEN GOTO 0500
0200 IF C$(1,3)="CHG" THEN GOTO 0720
0210 IF C$(1,3)="NEW" THEN GOTO 1340
0220 IF C$(1,3)="STP" THEN GOTO 1380
0230 GOTO 0060
0240 OPEN FILE (0,3),"BASIC.ID"
0250 INPUT FILE (0),S$
0260 IF EOF(0) THEN GOTO 0290
0270 PRINT S$
0280 GOTO 0250
0290 CLOSE FILE (0)
0300 GOTO 0140
0310 OPEN FILE (0,2),"BASIC.ID"
0320 INPUT "ACCOUNT NAME: ",A$
0330 IF LEN(A$)>=4 THEN GOTO 0360
0340 PRINT "ACCOUNT NAME MUST BE AT LEAST 4 CHARACTERS"
0350 GOTO 0320
0360 INPUT "PASSWORD (CR FOR NONE): ",P$
0370 IF LEN(P$)=0 THEN GOTO 0410
0380 IF LEN(P$)=4 THEN GOTO 0420
0390 PRINT "PASSWORD MUST BE 4 CHARACTERS"
0400 GOTO 0360
0410 LET P$=""
0420 INPUT "USER DIRECTORY (CR FOR NONE): ",D$
0430 IF LEN(D$)=0 THEN LET D$=""
0440 INPUT "AUTOLOGON FILE (CR FOR NONE): ",B$
0450 LET H5$="/",H$
0460 IF LEN(H$)=0 THEN LET H5$=""
0470 PRINT FILE (0),A$:"/":P$:"/":D$:H5$
0480 CLOSE FILE (0)
0490 GOTO 0140
0500 OPEN FILE (0,3),"BASIC.ID"
0510 OPEN FILE (1,1),"$BASICID.TM"
0520 INPUT "ACCOUNT NAME OF ENTRY YOU WISH TO DELETE: ",A$
0530 IF LEN(A$)>=4 THEN GOTO 0560
0540 PRINT "ACCOUNT NAME MUST BE AT LEAST 4 CHARACTERS"
0550 GOTO 0520
0560 INPUT FILE (0),S$
0570 IF EOF(0) THEN GOTO 0610
0580 IF S$(1,LEN(A$))=A$ THEN GOTO 0660
0590 PRINT FILE (1),S$
0600 GOTO 0560
0610 CLOSE FILE (0)
0620 CLOSE FILE (1)
0630 DELETE "BASIC.ID"
0640 RENAME "$BASICID.TM","BASIC.ID"
0650 GOTO 0140
0660 PRINT
0670 PRINT S$
0680 INPUT "IS THIS THE ENTRY YOU WISH TO DELTE ? ",Q$
0690 IF LEN(Q$)=0 THEN GOTO 0560
0700 IF Q$(1,1)="Y" THEN GOTO 0560
0710 GOTO 0590
0720 OPEN FILE (0,3),"BASIC.ID"
0730 OPEN FILE (1,1),"$BASIC.TM"
0740 INPUT "ACCOUNT NAME OF ENTRY YOU WISH TO CHANGE: ",A$
0750 IF LEN(A$)>=4 THEN GOTO 0780
0760 PRINT "ACCOUNT NAME MUST BE AT LEAST 4 CHARACTERS"
0770 GOTO 0740
0780 INPUT FILE (0),S$
0790 IF EOF(0) THEN GOTO 0830
0800 IF S$(1,LEN(A$))=A$ THEN GOTO 0870
0810 PRINT FILE (1),S$
0820 GOTO 0780
0830 CLOSE
0840 DELETE "BASIC.ID"
0850 RENAME "$BASICID.TM","BASIC.ID"
0860 GOTO 0140
0870 LET N4=""
0880 LET P$=""
0890 LET D$=""
0900 LET H$=""
0910 LET J=1
0920 FOR I=1 TO LEN(S$)
0930 IF S$(I,I)="/" THEN GOTO 0970
0940 ON J THEN GOSUB 0990, 1010, 1030, 1050
0950 NEXT I
0960 GOTO 1070
0970 LET J=J+1
0980 GOTO 0950
0990 LET N$=N$,S$(I,I)
1000 RETURN
1010 LET P$=P$,S$(I,I)
1020 RETURN
1030 LET D$=D$,S$(I,I)
1040 RETURN
1050 LET H$=H$,S$(I,I)
1060 RETURN
1070 PRINT "CHANGE ACCOUNTNAME [":N$:"]":
1080 INPUT " (Y OR CR)? ",Q$
1090 IF LEN(Q$)=0 THEN GOTO 1140
1100 INPUT "NEW ACCOUNT NAME: ",N$
1110 IF LEN(N$)>=4 THEN GOTO 1140
1120 PRINT "ACCOUNT NAME MUST BE AT LEAST 4 CHARACTERS"
1130 GOTO 1100
1140 PRINT "CHANGE PASSWORD [":P$:"]"
1150 INPUT " (Y OR CR)? ",Q$
1160 IF LEN(Q$)=0 THEN GOTO 1210

```

Figure 3-1. Program EDITID.QA (continues)

```

1170 INPUT "NEW PASSWORD: ",P$
1180 IF LEN(P$)=4 THEN GOTO 1210
1190 PRINT "PASSWORD MUST BE 4 CHARACTERS"
1200 GOTO 1170
1210 PRINT "CHANGE DIRECTORY [":D$:"]"
1220 INPUT " (Y OR CR)? ".Q$
1230 IF LEN(Q$)=0 THEN GOTO 1260
1240 INPUT "NEW DIRECTORY FOR USER: ",D$
1250 IF LEN(D$)=0 THEN LET D$=""
1260 PRINT "CHANGE AUTOLOGON FILE [":H$:"]";
1270 INPUT " (Y OR CR)? ".Q$
1280 IF LEN(Q$)=0 THEN GOTO 1300
1290 INPUT "NAME OF AUTOLOGON FILE: ".H$

1300 LET H$="/",H$
1310 IF LEN(H$)=0 THEN LET H$=""
1320 PRINT FILE (1),N$;"/";P$;"/";D$;H$
1330 GOTO 0780
1340 DELETE "BASIC.ID"
1350 OPEN FILE (0,1),"BASIC.ID"
1360 CLOSE FILE (0)
1370 GOTO 0140
1380 PRINT
1390 DIR "BASIC"
1400 PRINT "YOU ARE NOW IN BASIC.DR"
1410 ESC
1420 NEW

```

Figure 3-1. Program EDITID.QA (concluded)

```

* RUN
WHICH DIRECTORY IS BASIC.ID IN ? DP4

COMMANDS ARE:
DIS - TO DISPLAY BASIC.ID
ADD - TO ADD AN ENTRY TO BASIC.ID
DEL - TO DELETE AN ENTRY FROM BASIC.ID
CHG - TO CHANGE AN ENTRY IN BASIC.ID
NEW - TO DELETE BASIC.ID
STP - STOP

COMMAND ? DIS
0000//USER0/SWAPTIMESP.CI
0001//USER1/SWAPTIMESP.CI
0002//USER2/SWAPTIMESP.CI
0003//USER3/SWAPTIMESP.CI
0004//USER4
0005//USER5
0006//USER6
0007//USER7
0008//USER8
MAST//BASIC
PLAY//GAMES
TEST//QATESTS/TESTA.CI
IHOR/1978/GAMES
UWOO/UWASH5/DRIVER
DISK//
HUNT//HUNTINGTON
9990//USER4/HELLO
9991//USER4/HELLO
9992//USER4/HELLO
9993//USER4/BING
9994//USER4/BING
9995//USER4/BING

COMMAND ? CHG

ACCOUNT NAME OF ENTRY YOU WISH TO CHANGE: IHOR
CHANGE ACCOUNT NAME [IHOR] (Y OR CR)?
CHANGE PASSWORD [1978] (Y OR CR)? Y
NEW PASSWORD: 1954

CHANGE DIRECTORY [GAMES] (Y OR CR)? Y
NEW DIRECTORY FOR USER: BASIC
CHANGE AUTOLOGON FILE [] (Y OR CR)?

COMMAND ? DEL
ACCOUNT NAME OF ENTRY YOU WISH TO DELETE: DISK
DISK//
IS THIS THE ENTRY YOU WISH TO DELETE ? Y
COMMAND ? ADD
ACCOUNT NAME: BILL
PASSWORD (CR FOR NONE): NOVA
USER DIRECTORY (CR FOR NONE): BASIC
AUTOLOGON FILE (CR FOR NONE): HELLO.CI

COMMAND ? DIS
0000//USER0/SWAPTIMESP.CI
0001//USER1/SWAPTIMESP.CI
0002//USER2/SWAPTIMESP.CI
0003//USER3/SWAPTIMESP.CI
0004//USER4
0005//USER5
0006//USER6
0007//USER7
0008//USER8
MAST//BASIC
PLAY//GAMES
TEST//QATESTS/TESTA.CI
IHOR/1954/BASIC
UWOO//UWASH5/DRIVER
HUNT//HUNTINGTON
9990//USER4/HELLO
9991//USER4/HELLO
9992//USER4/HELLO
9993//USER4/BING
9994//USER4/BING
9995//USER4/BING
BILL/NOVA/BASIC/HELLO.CI

COMMAND ? STP
YOU ARE NOW IN BASIC.DR

```

Figure 3-2. Results of Program EDITID.QA

```

* NEW "USERS.QA"
* LIST
0010 PRINT "<12>"
0020 PRINT "<8>"
0030 LET S=SYS(17)/10
0040 LET H=INT(S/3600)
0050 LET S=S-3600*H
0060 LET M=INT(S/60)
0070 LET S=S-60*M
0080 LET S=(INT(10*S+.5))/10
0090 PRINT "THE SYSTEM HAS BEEN UP";H;"HOURS,";M;"MINUTES,";
      S;"SECONDS";"<12>"
0100 USERS
0110 GOTO 0020

```

Figure 3-3. Program USERS.QA

```

* RUN

THE SYSTEM HAS BEEN UP 0 HOURS, 6 MINUTES, 14.6 SECONDS
NAME STATUS DIRECTORY LINE SWPS DATA PROG PRI OVLY FILES CPU
0000 0001 USER0 0 10 7 5 12 1 25 -1 0 19.8
0001 0001 USER1 1 12 0 3 3 1 25 -1 0 32.3
IHOR 0001 BASIC 2 13 23 1 24 1 25 -1 0 2.6
0000 3
0004 0001 USER4 4 12 20 2 22 1 25 -1 0 52.6
BILL 0005 BASIC 5 11 ... 11 ... 1 25 -1 0 6.9
0000 6
0000 7
0000 8
0000 16
OPER 0001 BASIC -1 23 0 1 1 1 25 9 1 6.2

THE SYSTEM HAS BEEN UP 0 HOURS, 6 MINUTES, 21 SECONDS
NAME STATUS DIRECTORY LINE SWPS DATA PROG PRI OVLY FILES CPU
0000 0001 USER0 0 13 7 5 12 1 25 -1 0 20.6
0001 0001 USER1 1 16 9 3 12 1 25 -1 0 33
IHOR 0001 BASIC 2 17 0 1 1 1 25 -1 0 3.7
0000 3
0004 0001 USER4 4 16 16 2 18 1 25 -1 0 53.3
BILL 0001 BASIC 5 14 4 11 15 1 25 -1 0 7.8
0000 6
0000 7
0000 8
0000 16
OPER 0001 BASIC -1 29 1 1 2 1 25 9 1 6.5

THE SYSTEM HAS BEEN UP 0 HOURS, 6 MINUTES, 24.1 SECONDS
NAME STATUS DIRECTORY LINE SWPS DATA PROG PRI OVLY FILES CPU
0000 0001 USER0 0 15 15 5 20 1 25 -1 0 21.1
0001 0001 USER1 1 17 21 3 24 1 25 -1 0 33.7
IHOR 0005 BASIC 2 19 ... 1 ... 1 25 -1 0 4.2
0000 3
0004 0001 USER4 4 19 22 2 24 1 25 -1 0 53.9
BILL 0001 BASIC 5 16 3 11 14 1 25 -1 0 8.5
0000 6
0000 7
0000 8
0000 16
OPER 0001 BASIC -1 31 7 1 8 1 25 9 1 6.8

STOP AT 0050

```

Figure 3-4. Results of Program USERS.QA

```

* NEW "FREEID.QA"
* LIST
0010 DIM A$(30), B$(30)
0020 INPUT "WHICH DIRECTORY IS BASIC.ID IN? ", A$
0030 DIR A$
0040 OPEN FILE (1,3), "BASIC.ID"
0050 OPEN FILE (2,1), "FREEID.TM"
0060 INPUT FILE (1), A$
0070 IF EOF(1) THEN GOTO 0170
0080 LET I=I+1
0090 LET A=POS(A$, "/ ", 1)
0100 LET A$=A$(1, A-1)
0110 LET A1$="100"
0120 LET B=LEN(STR$(I))
0130 LET A1$=A1$(1, 4-B)
0140 LET B$=A1$, STR$(I), " FREE ", A$
0150 PRINT FILE (2), B$
0160 GOTO 0060
0170 CLOSE FILE (1)
0180 CLOSE FILE (2)
0190 ENTER "FREEID.TM"
0200 DELETE "FREEID.TM"
0210 ON ERR THEN GOSUB 2020
0220 DIR "BASIC"
2000 PRINT "ALL LINES <34>FREE<34>ED"
2010 END
2020 IF SYS(7)=51 THEN RETURN
2030 PRINT "ERROR #";SYS(7);" AT LINE #";SYS(19);
    " HALTED PROGRAM!!!"
2040 STOP

```

Figure 3-5. Program FREEID.QA

```

* USERS
NAME STATUS DIRECTORY LINE SWPS DATA PROG PRI OVLY FILES CPU
0000 0005 USER0 0 34 ... 5 ... 1 25 -1 0 26.0
0001 0001 USER1 1 39 0 3 3 1 25 -1 0 38.6
IHOR 0001 BASIC 2 43 6 1 7 1 25 -1 0 9.9
0000 3
0004 0001 USER4 4 40 22 2 24 1 25 -1 0 58.9
BILL 0001 BASIC 5 37 10 11 21 1 25 -1 0 14.2
0000 6
0000 7
0000 8
0000 16
OPER 0000 BASIC -1 36 8 1 9 1 25 9 1 7.1
* RUN
WHICH DIRECTORY IS BASIC.ID IN? DP4
ALL LINES "FREE"ED

END AT 2010
* USERS
NAME STATUS DIRECTORY LINE SWPS DATA PROG PRI OVLY FILES CPU
0000 1000 USER0 0 42 0 5 5 1 25 -1 0 27.4
0001 1004 USER1 1 46 ... 3 ... 1 25 -1 0 39.7
IHOR 1004 BASIC 2 50 ... 1 ... 1 25 -1 0 10.9
0000 3
0004 1000 USER4 4 46 18 2 20 1 25 -1 0 59.7
BILL 1000 BASIC 5 43 6 11 17 1 25 -1 0 15.5
0000 6
0000 7
0000 8
0000 16
OPER 0000 BASIC -1 42 22 1 23 2 25 9 1 7.8
* USERS
NAME STATUS DIRECTORY LINE SWPS DATA PROG PRI OVLY FILES CPU
0000 1000 USER0 0 42 0 5 5 1 25 -1 0 27.4
0001 1004 USER1 1 46 ... 3 ... 1 25 -1 0 39.7
IHOR 1004 BASIC 2 50 ... 1 ... 1 25 -1 0 10.9
0000 3
0004 1000 USER4 4 46 18 2 20 1 25 -1 0 59.7
BILL 1000 BASIC 5 43 6 11 17 1 25 -1 0 15.5
0000 6
0000 7
0000 8
0000 16
OPER 0000 BASIC -1 42 22 1 23 2 25 9 1 8.3

```

Figure 3-6. Results of Program FREEID.QA


```

* NEW "KILLID.QA"
* LIST
0010 DIM A$(30),B$(30)
0020 INPUT "WHICH DIRECTORY IS BASIC.ID IN? ",A$
0030 DIR A$
0040 OPEN FILE (1,3),"BASIC.ID"
0050 OPEN FILE (2,1),"KILLID.TM"
0060 INPUT FILE (1),A$
0070 IF EOF(1) THEN GOTO 0170
0080 LET I=I+1
0090 LET A=POS(A$,"/",1)
0100 LET A$=A$(1,A-1)
0110 LET A1$="100"
0120 LET B=LEN(STR$(I))
0130 LET A1$=A1$(1,A-B)
0140 LET B$=A1$,STR$(I)," KILL ",A$
0150 PRINT FILE (2),B$
0160 GOTO 0060
0170 CLOSE FILE (1)
0180 CLOSE FILE (2)
0190 ENTER "KILLID.TM"
0200 DELETE "KILLID.TM"
0210 ON ERR THEN GOSUB 2030
0220 DIR "BASIC"
2000 PRINT "ALL LINES <34>KILL<34>ED, DELAYING UNTIL
      TERMINAL I/O IS COMPLETE"
2010 ON ERR THEN GOTO 2060
2020 BYE
2030 IF SYS(7)=51 THEN RETURN
2040 PRINT "ERROR #";SYS(7);" AT LINE #";SYS(19);
      " HALTED PROGRAM!!!"
2050 STOP
2060 IF SYS(7)=64 THEN RETRY
2070 GOTO 2040

```

Figure 3-7. Program KILLID.QA

```

* RUN
WHICH DIRECTORY IS BASIC.ID IN? DP4
ALL LINES "KILL"ED, DELAYING UNTIL TERMINAL I/O IS COMPLETE
OPER 08/23/83 14:09 SIGN OFF, -1
OPER 08/23/83 14:09 CPU USED, 10.8
OPER 08/23/83 14:09 I/O USED, 347, 22

```

Figure 3-8. Results of Program KILLID.QA

Notice in Figure 3-6 that the USERS command first is issued to see who is on the system, and then again to show the results of the FREEID.QA program.

Figure 3-7 shows the program KILLID.QA, which allows you to force users off the system. The program checks BASIC.ID to get the user profiles, then kills every profile, including yours.

The results in Figure 3-8 show that KILLID.QA logs everybody off, including itself and the operator. The BYE command in the program outputs the final accounting total.

System Performance Considerations

In a multiuser system, disk swapping degrades performance and therefore normally should be avoided. Determining the size of the largest program that may be run without disk swapping is an important system management task. The size varies from installation to installation, depending upon the number of terminals and the amount of memory on the system, and possibly on other factors. Swapping occurs whenever the total size of all user programs exceeds the user program window of the RDOS ground in which Extended BASIC is running — 32K words (64KB) for ECLIPSE and 31K words (62KB) for NOVA computers. In unmapped systems, swapping is always to disk. In mapped systems, equipped with a Memory Management and Protection Unit, swapping is first to extended memory; disk swapping occurs only if all loaded user programs exceed the extended memory size of the Extended BASIC ground. Keep in mind, however, that maximum program size shrinks as more lines are added to a system. In multiuser systems, the actual space available for user programs without swapping is less than these maximum sizes.

For example, an RDOS system with 128K words of memory might be organized as illustrated in Figure 3-9:

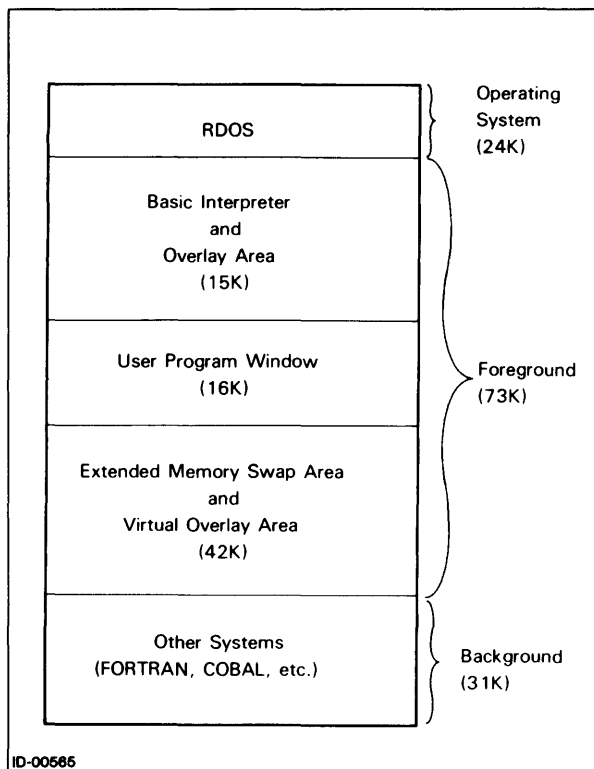


Figure 3-9. Memory Allocation in Words, in an RDOS System

This system could run four BASIC programs of 14K words (28KB) simultaneously without disk swapping. Under this system, the largest BASIC program that could be run would be 16K words (32KB).

Extended memory swapping is much faster than disk swapping, but it still exacts a cost in additional internal overhead. If you increase the Extended BASIC ground above 32K words, you should increase it substantially so that the increased swapping area compensates for the additional overhead.

Use the CLI SMEM command to set the size of the Extended BASIC ground up to the maximum of 256K words (512KB). For more information on manipulating the background/foreground size, see the *RDOS/DOS Command Line Interpreter Reference Manual*.

System Security Considerations

In addition to maintaining the BASIC timesharing system, the system manager must safeguard the security and integrity of user programs and data files. Data General's Extended BASIC provides the capabilities by which you can:

- Restrict access to the system by any potential user

- Control directory sharing
- Monitor and control the use of system facilities
- Restrict access to program and data files

Controlling Access to the System

The BASIC.ID identification file controls all access to the BASIC system. This file contains the user identification code, password, directory assignment, and logon program for each system user. (The file is described in chapter 1.) The entries you make to this file define a complete, exclusive set of system users.

No one can use the system without knowing a user identification code listed in the BASIC.ID file, and only one user at a time may use that code. You can create, delete, or modify entries in this file as often as necessary.

As each user logs on, BASIC outputs the ACCOUNT NAME prompt at the user's terminal.

The user identification code is not necessarily secure information, since it is required as an argument to the MSG communication command, appears in any listing of the *name.AF* accounting file, and is displayed on the system console in response to a USERS command. Therefore, BASIC allows you to add an optional password to each user entry in the BASIC.ID file. This password is an extension of up to four characters to the user identification code. The user and the system manager select it, and it appears only in the BASIC.ID file.

After you enter a user's password in the BASIC file, BASIC issues a PASSWORD prompt whenever that user logs on. The password itself is never echoed on the terminal. Three incorrect answers to the prompt place that user's terminal back in the ready state, where it awaits another ESC or RUBOUT.

If you (or a user) suspect that an unauthorized person has learned a password, you should change the BASIC.ID file promptly.

You should keep the BASIC.ID file itself secure from examination or modification by anyone else. You should not keep this file in any directory accessible to system users (such as BASIC.DR), nor should you ever give it the O sharable attribute via CHATR. You could use CHATR to make the file write-protected (W) and permanent (P); however, you would have to remove these attributes whenever you modified the file.

Directory Sharing

Although you must assign a unique identification code to each user, several users may share the same directory or partition. However, approach directory sharing with extreme caution for two reasons:

- If two users share a directory, each can freely use the other's files and can inadvertently destroy them by using the same filename in a LIST or SAVE command.
- Data files can be inadvertently overwritten the same way.

Monitoring the System

In addition to the cumulative record of system usage maintained in the accounting file (*name.AF*), you can get an immediate report on active users with the USERS command. With this report and the communication commands MSG, FMSG, ALL, and FALL, you can determine user activity and obtain desired information from a user.

If an unauthorized person is using someone else's logon identification, there are several options. The MSG and FMSG commands enable you to establish a dialog with the offending user. The FREE and KILL commands can halt that user's program and force the user off the system. After the offending user has logged off, you can change the compromised identification code in the BASIC.ID file.

Protecting Programs and Data Files

Normally, users have complete access to all the files in their own directories, and limited access to files in public directories (BASIC.DR) and other users' directories. You can access any user's directory from the master terminal by naming that directory in a DIR command.

The BASIC system can make any user's files accessible by other users on a selective basis, or restrict the type of access even for the owner of a file. The CHATR command provides this capability. Attributes that the BASIC CHATR command can set are compatible with those of the CLI CHATR command.

You can examine the attributes of any file by using the BASIC WHATS command, or the CLI command LIST/E. File attributes of particular importance in maintaining file security are:

- P Permanent file. You cannot delete the file unless the attribute is removed.
- R Read protected. You cannot open the file for reading.
- W Write protected. You cannot open the file for writing.
- O Special BASIC argument to CHATR that makes a file sharable with attribute H. Other users may access the file if they know the name of the directory in which the file resides and the name of the file.
- E Special BASIC argument to CHATR that makes a file an execute-only file with attribute E. You may load or enter and run a program, but you cannot list, save, or modify it. The special attribute in the CLI is &.
- H Special BASIC argument to CHATR that specifies attributes H, P, and W. The special attribute in the CLI is ?.
- Special BASIC argument to CHATR that preserves current file attributes and adds those specified. Use the asterisk only in conjunction with other attributes in the argument.
- + Preserves current file attributes and adds those following the plus sign. Identical to the asterisk.
- Only those attributes following the minus sign are removed.
- 0 Removes current file attributes and adds the attributes to the right of the zero. For example, 0RW removes all current file attributes and replaces them with RW.

File attributes apply to all users, including the owner, and can be changed only by the owner or the system manager. It may be necessary to temporarily change the attributes of a file before it can be used. For example, a user might intentionally specify attributes PW to prevent the inadvertent deletion or overwriting of a file. The W attribute would have to be removed with the CHATR command before the file could be changed.

For one user to access a file in another user's directory, the following conditions must be met:

- The user must know the name of the appropriate directory and file.
- The file must have the O sharable attribute.
- The mode of access to the file must not conflict with any other attributes set for the file.

Batch Operation

BATCH is a program that allows system software to manage the processing of a job stream without operator intervention. It is not available to DOS users. When Extended BASIC runs under BATCH, the BASIC system becomes a single-user system dedicated to BATCH at the system terminal. This is true even for multiuser configurations. Any attempt to log on while the system is running under BATCH evokes the message:

BATCH

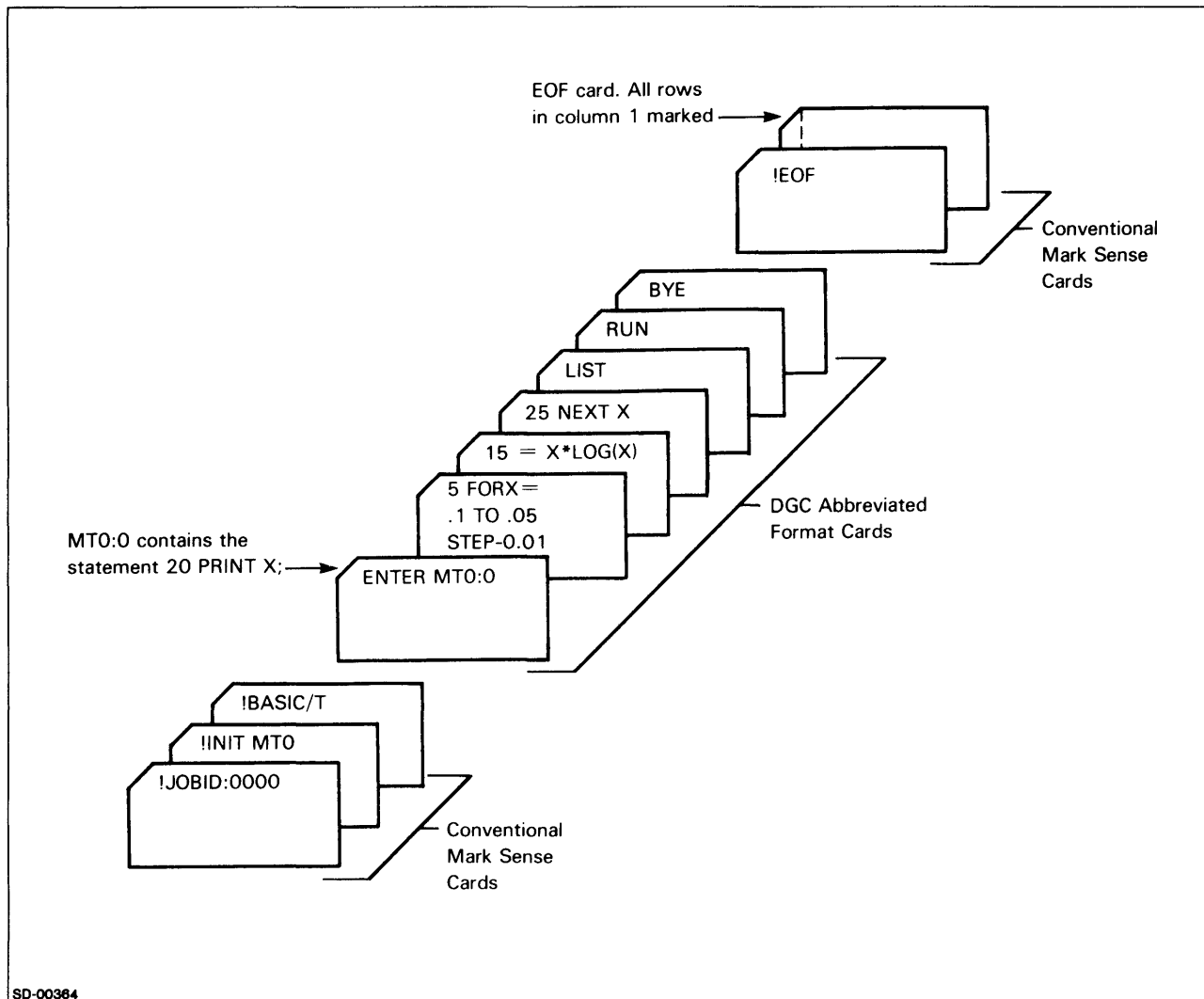


Figure 3-10. Sample BATCH BASIC Program

However, if the RDOS system is configured for background and foreground, then a multiuser BASIC and a BASIC under BATCH may run concurrently if there is no multiplexor conflict.

BATCH commands begin with an exclamation point, as in the BATCH command to execute BASIC:

- R (CLI prompt)
- BATCH (Execute BATCH)
- !JOB ID:xxxx (xxxx is the card that contains the first BATCH command in the job; it might include the user ID.)
- !save-file The name of the BASIC system save (.SV) file specified during BSG

When using BASIC under BATCH, the user writes the same series of commands and statements into the job stream as would be input to the terminal.

BATCH requires a log file, an input file, and an output file. The default assignment for the log file is the terminal; the default device for the input file (called SYSIN) is the card reader (\$CDR); and the default device for the output file (called SYSOUT) is the line printer (\$LPT). These default files may be changed by the user, as described in the *BATCH User's Manual* (093-000087).

Extended BASIC commands under BATCH must not attempt any input/output to unit record devices assigned to SYSIN or SYSOUT. For example, the following are illegal commands if \$CDR is SYSIN or \$LPT is SYSOUT:

```
CARDS"$ACDR"  
LIST "$LPT"
```

If your system has a mark-sense card reader, and you plan to use DGC abbreviated format cards, you must include the /T switch after the name of the BASIC system save file:

```
R  
BATCH  
!JOB ID:xxxx  
!save-file/T  
.  
.  
.  
!EOF
```

The /T switch indicates that each command or statement in the SYSIN (\$CDR) job stream after you invoke BASIC is in abbreviated format and is to be translated by BASIC.

If SYSIN is not a card reader, and you use a CARDS \$CDR (with the abbreviated format cards), then you do not need the /T switch to specify translation for the cards. However, the SYSIN file must be in standard BASIC format. CARD \$CDR is the only BASIC statement that recognizes the DGC abbreviated format mark-sense cards when you have not set the /T switch.

You can use the mark-sense card reader as SYSIN (using abbreviated format cards and specifying the /T switch), and you can execute a CARDS command or statement for a disk file that is not in abbreviated format. The /T refers only to card translation.

Note that a mark-sense card reader does not necessarily require abbreviated format cards; you can use conventional cards as well. The /T switch is not required with conventional cards.

In the BSG dialog, the query

```
MARK SENSE CARD READER (O,M,NO)?
```

asks whether or not the card reader is to be used for abbreviated format cards. If it is, specify Overlay (O) or Main (M); otherwise, specify NO.

The example of BATCH BASIC in Figure 3-10 uses a mark-sense card reader as SYSIN. Note that the commands in the job stream (\$CDR) are prepared on standard cards in Hollerith code. When you execute BASIC/T from the job stream, all subsequent commands and statements are prepared on cards in the DGC abbreviated format.

End of Chapter

Appendix A

Initialization Error Messages

Extended BASIC error messages are printed as two-digit codes, followed by a brief explanatory message. Errors are of four kinds: syntax, runtime, file I/O, and initialization. Table A-1 summarizes initialization error messages. See the *Extended BASIC Reference* for a comparable summary of the other three types of error messages.

Table A-1. Initialization Errors

Error Message	Explanation
ERROR 01-REAL-TIME CLOCK	Real time clock is not in the operating system.
ERROR 02-OVERLAY FILE	The overlay file (<i>name.OL</i>) for the invoked BASIC system cannot be opened. It may have been deleted, altered or renamed. In any case, it is necessary to generate another BASIC system as described in chapter 1. A disk data error may be the cause of these errors. This requires rebuilding the RDOS system and restoring BASIC system files as described in chapter 1.
ERROR 03-INSUFFICIENT CHANNELS	Your RDOS system does not have enough channels to handle the maximum number of files requested by your BASIC system. Regenerate RDOS for more channels.
ERROR 04-INSUFFICIENT FREE STORAGE	<i>Name.CL</i> may have been altered.
ERROR 05-SWAPPING FILE	A BASIC system, configured for swapping, cannot open file BASIC.SW. The file may be in use or there may not be enough contiguous disk space available to break BASIC.SW. Try the RDOS CLEAR command to reset the file's use count to 0. If that does not work, it may be necessary to generate a BASIC system that allocates each user less memory space. Another alternative is to dump the files on the swapping device to magnetic tape for back-up and then selectively delete files from the swapping device to create more file space.
ERROR 06-INSUFFICIENT TCB's	This is an unexpected system software error. It can be caused if the BASIC.CL file, created by the BASIC system generation program, is altered before it is used to generate the system (i.e., @BASIC.CL@). If this is not the case, contact your local Data General representative.
ERROR 07-MULTIPLEXOR	BASIC uses the RDOS multiplexor handlers for supporting the 4060 and ALM multiplexors. Therefore, the current RDOS system was generated without a QTY or ALM device in the system. Generate a new RDOS system that includes QTY or ALM support. Or, you may have selected line numbers in your BASIC generation (BSG) that do not physically exist. In this case, regenerate BASIC, choosing the proper multiplexor line numbers.
ERROR 08-MASTER DIRECTORY	BASIC.DR is not in the system. BASIC.DR does not exist, or it is not initialized, or it cannot be found to be initialized by BASIC. BASIC requires a directory named BASIC.DR for use as a public directory. This directory contains the files that are listed when the BASIC command LIBRARY is given. Use the RDOS CDIR or CPART commands to create a subdirectory or secondary partition in the master device primary partition. Or, if BASIC.DR does exist, initialize it and then execute BASIC again.
ERROR 09-ID FILE	There is no BASIC.ID file in the system, or the BASIC.ID file's use count is not zero. Either create BASIC.ID as described in chapter 1, or use the RDOS CLEAR command to reset the file's use code to 0.
ERROR 10-ACCOUNTING FILE	BASIC.AF's use count is not zero. Use the RDOS CLEAR command to reset the file's use code to 0.
ERROR 12-EXTENDED MEMORY	There is not enough extended memory available to load the virtual overlay file. Use the RDOS SMEM command to increase the available memory.
NO ROOM FOR UFS's:filename	An RDOS system system error: the number of channels requested by BASIC is more than the number available from the operating system. Either generate a new operating system with more channels, or generate a new BASIC that requires fewer channels. BSG query 17 (chapter 1) explains the method for calculating system channel requirements for a BASIC system.

End of Appendix

Appendix B

Statement, Command, and Function Summary

The following table lists alphabetically, and summarizes briefly, the statements, commands, and functions of Extended BASIC. The RDOS and DOS privileged commands are indicated with a P in the column for the

operating system. These commands can be executed only at the system terminal.

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
ABS(expr) The absolute value of an expression.			•	•	•
ACL "filename" [, "UserID", "attributes"]... Prints a report of, or changes, the access control list for a file.	•	•		•	
ALL message Sends a message to all active users.	•	•			P •
ATN(expr) The arctangent of an angle. Result expressed in radians.			•	•	•
AUDIT ["filename"] Audits your terminal's input and output.	•	•		•	•
BYE Sign-off command.	•	•		•	•
CALL subr [,expr]... Calls an assembly language subroutine.	•	•		•	•
CARDS "filename" Enters program in mark-sense card format.	•	•			•
CDIR name Creates a subdirectory.	•	•			P •
CHAIN "filename" [THEN GOTO line no.] Transfers control to the program named in the statement.	•	•		•	•
CHAR $\left[\begin{array}{l} \left\{ \begin{array}{l} \text{"ON"} \\ \text{"OFF"} \\ \text{"characteristics"} \\ \text{"LPP", svar} \\ \text{"CPL", svar} \\ \text{"device"} \end{array} \right\} \left\{ \begin{array}{l} \text{"ON"} \\ \text{"OFF"} \\ \text{"characteristics"} \\ \text{"LPP", svar} \\ \text{"CPL", svar} \\ \text{"device"} \end{array} \right\} \dots \end{array} \right]$ Prints a report of, or changes, the current device characteristics.	•	•		•	

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
CHAR $\left[\left[\left\{ \begin{array}{l} \text{"ON"} \\ \text{"OFF"} \end{array} \right\} \text{"characteristics'/.} \right] \dots \right]$ Prints a report of, or changes, the current device characteristics.	•	•			•
CHATR "filename", attributes Changes file attributes.	•	•			•
CHR\$(expr) Generates the character represented in the ASCII collating sequence by a number.			•	•	•
CLI [command] Provides access to the CLI without terminating the BASIC process.	•	•		•	
CLOSE $\left\{ \begin{array}{l} \text{FILE} \\ \# \end{array} \right\} (\text{file})$ Closes an open file or files.	•	•		•	•
CON Continues execution of a stopped program.		•		•	•
COS(expr) The cosine of an angle expressed in radians.			•	•	•
CPART name,size Creates a secondary partition.	•	•			P •
CPU(expr) Returns a value (0 or 1) equal to the status of a CPU console switch.			•		•
DATA $\left\{ \begin{array}{l} \text{val} \\ \text{"str lit"} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{val} \\ \text{"str lit"} \end{array} \right\} \dots \right]$ Defines data to be used by READ and MAT READ.	•			•	•
DEF Used with FNa(d) function to define a user function.	•			•	•
DELAY = expr Delays program execution for a specified amount of time.	•	•		•	•
DELETE "filename" Deletes a file from your directory.	•	•		•	•
DIM $\left\{ \begin{array}{l} \text{svar}(n) \\ \text{array}(m) \\ \text{array}(\text{row},\text{col}) \end{array} \right\} \left[\left\{ \begin{array}{l} \text{.svar}(n) \\ \text{.array}(m) \\ \text{.array}(\text{row},\text{col}) \end{array} \right\} \dots \right]$ Specifies the size of string variables and numeric arrays.	•	•		•	•
DIR [pathname] Displays, sets or stores the pathname of the current working directory.	•	•		•	
DIR $\left[\left\{ \begin{array}{l} \text{primary part} [[:\text{secondary part} [:\text{subdirectory}]] \\ \text{secondary part} [:\text{subdirectory}] \\ \text{subdirectory} \\ \text{null-length string variable} \end{array} \right\} \right]$ Changes the directory and prints the name of the current directory.	•	•			P •
DISABLE Prevents the use of system console breaks.	•	•			P •

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
DISK Prints the number of blocks used and the number available in the partition in which your directory resides.	•	•		•	•
ECHO Enables the display of characters on input.	•	•		•	•
ENABLE Cancels a DISABLE command.	•	•			P •
END Stops program execution.	•			•	•
ENTER { "filename" svar } Merges the program named into the current program.	•	•		•	•
EOF (file) Returns a 1 if an end of file is detected, otherwise, 0.			•	•	•
ERASE n1, n2 Deletes statements from a program.	•	•		•	•
ESC Enables an ESC interrupt condition.	•	•		•	•
EXP(expr) The value of e to the power of an expression.			•	•	•
FALL message Forces a message to all active users.	•	•			P •
{ FILE } ["template"] { # } Prints the names of files in your directory that match the template.	•	•		•	
{ FILE } { # } Prints the filenames in your directory.	•	•			•
FMSG userID message Forces a message to a specific user.	•	•			P •
FNa(d) A user function which is defined in a DEF statement and returns a value.			•	•	•
FOR control var = expr1 TO expr2 [STEP expr3] Begins a FOR-NEXT loop and defines the number of times the loop is executed.	•			•	•
FREE userID Interrupts execution of userID 's program.	•	•			P •

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
GDIR Verifies the name of your directory.	•	•		•	•
GOSUB line no. Transfers program control to the first statement of a subroutine.	•			•	•
GOTO line no. Transfers program execution to a specified line.	•			•	•
GPOS { FILE } (file), var { # }	•	•		•	•
Determines the current file pointer position in an open file.					
HELP "verb" Displays information about each BASIC statement and command.	•	•		•	•
IF { rel-expr } { /THEN/ statement } { expr } { THEN line no. }	•	•		•	•
Executes a statement based on whether an expression is true or false.					
INIT name Initializes a directory or device for accessibility.	•	•			P •
INPUT ["str lit"], { var } { ,var } ...:/ { svar } { ,svar }	•	•		•	•
Accepts data for variables from the terminal.					
INPUT { FILE } { (file) } { var } { svar } ... { # } { (file,record) } { svar } { svar }	•	•		•	•
Reads data in ASCII from a sequential-access file.					
INT (expr) The largest integer not greater than the expression.	•	•		•	•
KILL userID Forces a specific user off the system.	•	•			P •
LEN (svar) Returns the number of characters currently assigned to a string variable.			•	•	•
[LET] { svar } = expr { var }	•	•		•	•
Assigns values or solutions to formulas to a variable.					
LEVEL /userID/ Monitors the priority constant for any user.	•	•			•
LIBRARY { "directory", "template" } { "directory" } { "template" }	•	•		•	
Prints the names of files in the BASIC library directory that match the template.					

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
LIBRARY Prints the filenames in the library directory.	•	•			•
LIST $\left\{ \begin{array}{l} \left\{ \begin{array}{l} \text{line } n1 \\ TO \end{array} \right\} \text{line } n2 \\ , \\ \text{line } n1 \left\{ \begin{array}{l} TO \\ , \end{array} \right\} \text{line } n2 \end{array} \right\}$ ["filename"] Outputs part or all of the current program to the terminal or other ASCII device.	•	•		•	•
LOAD "filename" Loads a previously saved program into the program storage area.		•		•	•
LOCK iden, "filename", start, record-size [,time] Gives exclusive access to a record in a file.	•	•		•	
LOG (expr) The natural logarithm of an expression.			•	•	•
LREAD ["str lit",] svar [,svar] Reads a string that is terminated by either a null, form feed, or carriage return (NEW LINE) from the terminal.	•	•		•	•
LREAD {FILE} { (file) } , svar [,svar1] # { (file, record) } Reads a string from a record in either a random- or sequential-access file that is terminated by either a null, form feed, or carriage return (NEW LINE).	•	•		•	•
LWRITE svar [,svar1] Writes a string to your terminal that is delimited by either a null, form feed, or carriage return (NEW LINE).	•	•		•	•
LWRITE {FILE} { (file) } , svar [,svar1] # { (file, record) } Writes a string to a record into either a random- or sequential-access file which will be terminated by either a null, form feed, or carriage return (NEW LINE).	•	•		•	•
MAT mvar1 = mvar2 Assigns the dimensions and values of mvar2 to mvar1.	•	•		•	•
MAT mvar1 = mvar2 {+} mvar - Performs matrix addition or subtraction.	•	•		•	•
MAT mvar1 = {mvar2} * mvar3 {(expr)} Multiplies a matrix by a numeric expression or another matrix.	•	•		•	•
MAT mvar = CON [(row,col)] Sets the value of each matrix element to 1.	•	•		•	•
MAT mvar = IDN [(row,col)] Sets the elements of the major diagonal of a matrix to 1s and all other elements to 0s.	•	•		•	•

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
MAT mvar1 = INV (mvar2) Performs matrix inversion.	•	•		•	•
MAT mvar1 = TRN (mvar2) Transposes matrix mvar2.	•	•		•	•
MAT mvar = ZER ([row,] col) Sets the value of each matrix element to 0.	•	•		•	•
MAT INPUT ["str lit"] mvar [(row,col)][,mvar[(row,col)]]... Specifies matrixes for which you enter data from the terminal when the statement is executed.	•	•		•	•
MAT INPUT { FILE } { (file) } ,mvar [,mvar]... # { (file,record) } Reads a record of matrix data in ASCII format from a file.	•	•		•	•
MAT PRINT mvar [{ , } mvar] ... [{ : }] Prints the contents of the specified matrixes.	•	•		•	•
MAT PRINT { FILE } { (file) } , mvar [{ : } mvar... { : }] # { (file,record) } Outputs matrix file data to an ASCII device.	•	•		•	•
MAT READ mvar [(row,col)][,mvar[(row,col)]]... Reads data into the specified matrixes from the data list defined by DATA statement(s).	•	•		•	•
MAT READ { FILE } { (file) } ,mvar [,mvar]... # { (file,record) } Reads matrix data in binary format from a file.	•	•		•	•
MAT TINPUT [(line no. [,time]).] ["str lit".] mvar [(row,] col)] [,mvar [(row,]col)]... Reads values from your terminal and assigns them to the elements of a matrix or list of matrixes, within a prescribed time.	•	•		•	•
MAT WRITE { FILE } { (file) } ,mvar [,mvar]... # { (file,record) } Writes matrix data in binary format to a file.	•	•		•	•
MAX [val] Sets the limit for a number of active users.	•	•			P •
MSG { pid "processname" } , "message" "console name" Transmits a message from your terminal to another programmer or to the system operator.	•	•		•	
MSG userID message Transmits messages to other users or the operator or cancels NOMSG.	•	•			•

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
NEW ["filename"] Clears your storage area.	•	•		•	•
NEXT control var The last statement in a FOR-NEXT loop; changes the value of the control variable.	•	•		•	•
NOECHO Disables character echoing on input.	•	•		•	•
NOESC Disables ESC interrupt handling.	•	•		•	•
NOMSG Prevents receipt of messages.	•	•		•	•
ON ERR { THEN line no. / THEN statement }	•			•	•
ON ESC { THEN line no. / THEN statement }	•			•	•
ON expr { GOTO } line no. [, line no.]... { GOSUB }	•			•	•
OPEN { FILE } (file,mode),"filename" [,recordsize [,filesize]] { # }	•	•		•	•
ORD(expr) Represents the ordinal position of a character in the ASCII collating sequence.			•	•	•
PAGE=expr Sets the right margin of the terminal.	•	•		•	•
POS { (svar1) } { (svar2) } , (expr) { ("str lit 1") } { ("str lit 2") }			•	•	•
{ PRINT } { { expr } { "str lit" } { ; } { svar } { TAB(n) } } ... { { ; } }	•	•		•	•
{ PRINT } { FILE } { (file) } { (file,record) } { expr var svar "str lit" } { { ; } { svar } { "str lit" } } ... { { ; } }	•	•		•	•
Prints specified data.					
Outputs data to an ASCII device.					

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
$\left\{ \begin{array}{l} \text{PRINT} \\ \text{FILE} \end{array} \right\} (\text{file}, [\text{record}]), \text{ USING format, expr} [, \text{expr}] \dots$ $\left\{ \begin{array}{l} ; \\ \# \end{array} \right\}$ Formats output to files.	•	•		•	•
PRINT USING format, expr $\left\{ \begin{array}{l} , \text{expr} \\ , \text{expr} \end{array} \right\} \dots$ Formats printed output.	•	•		•	•
PUNCH $\left[\begin{array}{l} \left\{ \begin{array}{l} \text{line } n1 \\ \text{TO} \end{array} \right\} \text{line } n2 \\ , \\ \text{line } n1 \left\{ \begin{array}{l} \text{TO} \\ , \end{array} \right\} \text{line } n2 \end{array} \right]$ Outputs part or all of the current program to the terminal punch.		•			•
RANDOMIZE Resets the random number generator.	•	•		•	•
READ $\left\{ \begin{array}{l} \text{var} \\ \text{svar} \end{array} \right\} \left[\begin{array}{l} \left\{ \begin{array}{l} , \text{var} \\ , \text{svar} \end{array} \right\} \dots \end{array} \right]$ Reads data from DATA statements.	•	•		•	•
READ $\left\{ \begin{array}{l} \text{FILE} \\ \# \end{array} \right\} \left\{ \begin{array}{l} (\text{file}) \\ (\text{file, record}) \end{array} \right\} \left\{ \begin{array}{l} \text{var} \\ \text{svar} \end{array} \right\} \left[\begin{array}{l} \left\{ \begin{array}{l} , \text{var} \\ , \text{svar} \end{array} \right\} \dots \end{array} \right]$... Reads data in binary format from a file.	•	•		•	•
RELEASE name Prevents further I/O access to a previously initialized directory or device.	•	•			P •
REM /message/ Inserts explanatory comments into a program.	•			•	•
RENAME "oldfilename", "newfilename" Renames files.	•	•		•	•
RENUMBER $\left[\begin{array}{l} \left\{ \begin{array}{l} \text{line } n1 \\ \text{STEP} \end{array} \right\} n2 \\ , \\ \text{line } n1 \left\{ \begin{array}{l} \text{STEP} \\ , \end{array} \right\} n2 \end{array} \right]$ Renumbers statements in the current program.		•		•	•
RESET $\left[\begin{array}{l} \left\{ \begin{array}{l} \text{FILE} \\ \# \end{array} \right\} (\text{file}) \end{array} \right]$ Positions the file pointer to the beginning of a file.	•	•		•	•
RESTORE [line no.] Moves the data element pointer to the beginning of a data list or DATA statement line.	•	•		•	•
RETRY Repeats the statement which caused an error.	•			•	•

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
RETURN Last statement of a subroutine; returns program control to statement following last GOSUB statement executed.	•			•	•
RND(expr) Random number n, such that $0 \leq n < 1$.			•	•	•
RUN $\left[\begin{array}{l} \{ \text{"filename"} \} \\ \{ \text{line no.} \} \end{array} \right]$ Executes a program either from the first line or from a specified line.		•		•	•
SAVE "filename" Writes the current program into your directory or to a device in binary format.	•	•		•	•
SEARCHLIST $\left\{ \begin{array}{l} \{ \text{/pathname} \} \\ \{ \text{pathname} \} \end{array} \right\}$ Displays, sets or stores the searchlist setting. The current searchlist specifies which directories, in addition to the working directory, are searched for file references other than DELETE.	•	•		•	
SGN(expr) The algebraic sign of an expression.			•	•	•
SHARE "filename" Adds the resolution file attribute H (sharable) to a file that already exists in your directory.	•	•			•
SIN(expr) The sine of an angle expressed in radians.			•	•	•
SIZE Provides program and data storage usage information.	•	•		•	•
SPOS $\left\{ \begin{array}{l} \{ \text{FILE} \} \text{ (file), } \text{expr} \\ \{ \# \} \end{array} \right\}$ Moves the file pointer to the byte position specified by expr.	•	•		•	•
SQR(expr) The square root of an expression.			•	•	•
STOP Stops program execution.	•			•	•
STR\$(expr) Converts a numeric expression to its string representation.			•	•	•
SYS(0) The time of day (seconds past midnight).			•	•	•
SYS(1) The day of the month.			•	•	•
SYS(2) The month of the year.			•	•	•

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
SYS(3) The year.			•	•	•
SYS(4) The multiplexor line number or terminal number (-1 for operator's terminal).			•	•	•
SYS(5) CPU time used, in seconds.			•	•	•
SYS(6) The number of file I/O statements executed.			•	•	•
SYS(7) The error code of the last runtime error.			•	•	•
SYS(8) The number of the file most recently opened.			•	•	•
SYS(9) Page size.			•	•	•
SYS(10) Tab size.			•	•	•
SYS(11) Hour of the day.			•	•	•
SYS(12) Minutes past last hour.			•	•	•
SYS(13) Seconds past last minute.			•	•	•
SYS(14) Seconds remaining on timed input.			•	•	•
SYS(15) The constant PI (3.14159).			•	•	•
SYS(16) The constant e (2.71828).			•	•	•
SYS(17) 1/10 second clock.			•	•	•
SYS(18) Total number of BASIC I/O calls.			•	•	•
SYS(19) Line number of the last error.			•	•	•
TAB=expr Sets the zone spacing for PRINT statements.	•	•		•	•

Formats and Descriptions	S	C	F	AOS AOS/VS	RDOS DOS
TAB(expr) Function used with PRINT for tabulating to a column.			•	•	•
TAN(expr) The tangent of an angle expressed in radians.			•	•	•
TIME=expr Establishes the time limit for timed input operation.	•	•		•	•
TINPUT [(line no./time),] ["str lit",] {var} {svar} ...[:] Sets a time limit for programmer response.	•	•		•	•
UNLOCK [iden] Frees locked areas for use.	•	•		•	
UNSHARE "filename" Removes the resolution file attribute H (sharable) from a file that already exists in your directory.	•	•			•
USERS [userID] Prints a status report of all active users.	•	•			•
VAL {(svar)} {"str lit"} Returns decimal representation of a string.			•	•	•
var = DET(X) Obtains the determinant of the last matrix inverted by an INV statement.	•	•		•	•
WHATS "filename" Prints attributes and other information relating to a file.	•	•		•	•
WHO [{processID}] {"processname"} Identifies others on the system or provides your own identification.	•	•		•	•
WRITE {FILE}{(file)} {#}{(file,record)} {expr var svar "str lit"} {expr var svar "str lit"} ... Writes data in binary format to a file.	•	•		•	•

End of Appendix

Appendix C

Sample RDOS BSG Dialog

Extended BASIC Revision XX.XX MM/DD/YY System Generation

Valid answers are in parentheses. Default values are always given first and may be selected by a null response (carriage return). Defaults are followed by a list of alternatives, any one of which may be selected by typing a matching response.

Object computer type (NOVA,ECLIPSE,ORIGINAL) ?
Object operating system (RDOS,RTOS,DOS) ?
System save file (BASIC, any disk file name) ?
Accounting (YES,NO) ?
Multi-user system (NO,YES) ? YES
Line configuration: 0-7
Master console line number (-1,0-31) ?
Is master console a CRT (YES,NO) ?

Central processor option support:
Dial-up lines (NO,ALM,4060)
Swapping (YES,NO)
Hardware multiply/divide (NO,YES) ?
Hardware floating point (NO,SING,DOUB) ?
Memory management/protection (NO,YES) ? YES

User written subroutines (NO,YES) ? YES
String arithmetic (NO,YES) ? YES
Portion as overlay (YES,NO) ?
Default features acceptable (YES,NO) ?
Default overlays acceptable (YES,NO) ? NO

Valid answers to the following are:

O=Include feature as an overlay
M=Include feature in main storage
NO=Do not include feature

Matrix arithmetic (O,M,NO) ?
Matrix input and output (O,M,NO) ?

Matrix inverse (O,M,NO) ?
PRINT USING (O,M,NO) ?
BYE (O,M) ?
CALL (O,M)
Mark sense card reader (O,M,NO) ?
LIST,PUNCH (O,M) ?
File commands (O,M,NO) ?
USERS command (O,M) ?
Operator commands (O,M) ?
NEW,AUDIT (O,M) ?
User initialization (O,M) ?
DELETE,RENAME,PAGE,RENUMBER,SIZE,INIT,RELEASE (O,M,NO) ?
Console characteristics (O,M,NO) ?
DISK and directory commands (O,M,NO) ?
BASIC edit and help commands (O,M,NO) ?
Error message text (O,M,NO)
I/O error message text (O,M,NO) ?
SIN,COS,TAN,ATN,SYS (O,M,NO) ?
LOG,EXP,^,SQR,RND (O,M,NO) ?
LEN,POS,STR\$,VAL,ORD,CHR\$ (M,O,NO) ?

RUN filename,CHAIN,SAVE,LOAD,logon (O,M,NO) ?

Above overlays virtual (YES,NO) ?
Overlay statement compiler (NO,YES) ? YES
As virtual overlay (YES,NO) ?

Maximum files to be open at one time (18,1-90) ?
File sharing (YES,NO) ?
FOR NEXT nesting limit (8,4-20) ?
GOSUB nesting limit (8,4-20) ?
User function nesting limit (6,0-10) ?
Maximum user core size (in bytes) ?
Prompt string (*-space-null, any 10 character string) ?
Time of day with prompt (NO,YES) ?

End of Appendix

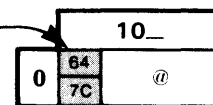
Appendix D

ASCII Character Set

To find the *octal* value of a character, locate the character, and combine the first two digits at the top of the character's column with the third digit in the far left column.

Legend:

Character code in decimal
 EBCDIC equivalent hexadecimal code
 Character



Octal	00_	01_	02_	03_	04_	05_	06_	07_
0	0 00 NUL	8 16 BS (BACK-SPACE)	16 10 DLE P	24 18 CAN X	32 40 SPACE	40 4D (48 F0 0	56 F8 8
1	1 01 SOH A	9 05 HT (TAB)	17 11 DC1 Q	25 19 EM Y	33 5A !	41 5D)	49 F1 1	57 F9 9
2	2 02 STX B	10 15 NL (NEW LINE)	18 12 DC2 R	26 3F SUB Z	34 7F " (QUOTE)	42 5C *	50 F2 2	58 7A :
3	3 03 ETX C	11 0B VT (VERT. TAB)	19 13 DC3 S	27 27 ESC (ESCAPE)	35 7B #	43 4E +	51 F3 3	59 5E ;
4	4 37 EOT D	12 0C FF (FORM FEED)	20 3C DC4 T	28 1C FS \ 	36 5B \$	44 6B , (COMMA)	52 F4 4	60 4C <
5	5 2D ENQ E	13 0D RT (RETURN)	21 3D NAK U	29 1D GS }	37 6C %	45 60 -	53 F5 5	61 7E =
6	6 2E ACK F	14 0E SO N	22 32 SYN V	30 1E RS }	38 50 &	46 4B . (PERIOD)	54 F6 6	62 6E >
7	7 2F BEL G	15 0F SI O	23 2E ETB W	31 1F US —	39 7D ' (APOS)	47 61 /	55 F7 7	63 6F ?

Octal	10_	11_	12_	13_	14_	15_	16_	17_
0	64 7C @	72 C8 H	80 D7 P	88 E7 X	96 79 \ (GRAVE)	104 88 h	112 97 p	120 A7 x
1	65 C1 A	73 C9 I	81 D8 Q	89 E8 Y	97 81 a	105 89 i	113 98 q	121 A8 y
2	66 C2 B	74 D1 J	82 D9 R	90 E9 Z	98 82 b	106 91 j	114 99 r	122 A9 z
3	67 C3 C	75 D2 K	83 E2 S	91 8D [99 83 c	107 92 k	115 A2 s	123 C0 }
4	68 C4 D	76 D3 L	84 E3 T	92 E0 \	100 84 d	108 93 l	116 A3 t	124 4F
5	69 C5 E	77 D4 M	85 E4 U	93 9D	101 85 e	109 94 m	117 A4 u	125 D0 }
6	70 C6 F	78 D5 N	86 E5 V	94 5F or ~	102 86 f	110 95 n	118 A5 v	126 A1 ~ (TILDE)
7	71 C7 G	79 D6 O	87 E6 W	95 6D — or _	103 87 g	111 96 o	119 A6 w	127 07 DEL (RUBOUT)

Character code in octal at top and left of charts.

| means CONTROL

ASCII Character Set

End of Appendix

Index

4

4060 1-4
4060,RTOS 1-13

A

access to BASIC, restrictions on RDOS and DOS 3-10
access to files, restrictions on RDOS and DOS 3-11
accounting file 1-3, 1-10, 3-11
AF file 1-3, 1-10, 3-11
allocating file space 1-8
ALM 1-2, 1-4
 RTOS 1-13
ASCII character set D-1
auto restart on RDOS 2-3

B

backup procedures 2-3
BASIC
 invoking on AOS and AOS/VS 1-17
 shutdown on RDOS and DOS 2-1
 testing on AOS and AOS/VS 1-18
BASIC configuration module 1-7
BASIC modules 1-2
BASIC system files
 AF file 1-3, 1-10, 3-11
 CL file 1-3, 1-7
 DL file 1-3, 1-7
 MP file 1-3, 1-8
 OL file 1-3, 1-7, 2-1
 RB file 1-3, 1-7
 SV file 1-3, 1-7, 2-1
 SW file 1-3
BASIC system generation
 RDOS and DOS 1-3
 RTOS 1-12
BASIC.DR 2-1
 creating 1-9
batch operation on RDOS 3-11
BSG 1-2, 1-3, 1-7, C-1
BSYSGEN 1-1
buffers, number of 1-1

C

cells 1-1
channels, maximum number of 1-1
CL file 1-3, 1-7
communicating with users, RDOS and DOS 3-1
core dump 1-2
crash procedures on RDOS and DOS 2-2
 auto restart 2-3
 updating records 2-2

D

device independence 3-1
dial-up lines 1-4
 RTOS 1-13
directories
 changing 3-1
 creating for BASIC, RDOS and DOS 1-8
 library, RDOS and DOS 1-9
 number of 1-1
 used in AOS and AOS/VS 1-16
 user, RDOS and DOS 1-9
directory sharing on RDOS and DOS 3-10
DL file 1-3, 1-7
double precision
 AOS and AOS/VS 1-15, 1-17
 RDOS and DOS 1-5
 RTOS 1-13

E

error messages
 during initialization 2-1, A-1
ESC 1-2

F

file sharing
 AOS and AOS/VS 1-17
 RDOS and DOS 1-7
filenames, RDOS and DOS

- BASIC root 1-3
- BASIC system 1-3
- files, RDOS and DOS
 - allocating space 1-8
 - creating for BASIC 1-8
 - maximum number open 1-6
- floating point
 - RDOS and DOS 1-4, 1-5
 - RTOS 1-13

G

- generating a system
 - dialog file, RDOS and DOS 1-7
 - RDOS or DOS 1-1
 - RTOS 1-11
- generating BASIC
 - RDOS and DOS 1-3
 - RDOS BSG dialog C-1
 - RTOS 1-11

I

- ID file, creating 1-9
- installing Extended BASIC
 - on AOS or AOS/VS 1-15
 - on RDOS or DOS 1-1
 - on RTOS 1-10
- interrupt character 1-2
- invoking Extended BASIC
 - AOS and AOS/VS 1-17
 - initialization 2-1
 - RDOS and DOS 2-1

L

- library directory, creating on RDOS and DOS 1-9
- lines
 - maximum number on RDOS and DOS 1-3
 - maximum number on RTOS 1-12
- load command line file, RDOS and DOS 1-7
- loading Extended BASIC, RDOS and DOS 1-2
- logon program, RDOS and DOS 1-9

M

- main memory, RDOS and DOS 1-5
- main storage features, RTOS 1-13
- map file, RDOS and DOS 1-8
- modules of BASIC
 - loading onto AOS and AOS/VS 1-16
 - loading onto RDOS and DOS 1-2

- MP file 1-3, 1-8
- multiply/divide
 - RDOS and DOS 1-5
 - RTOS 1-13
- multiuser system
 - generating on RDOS and DOS 1-3
 - generating on RTOS 1-12
 - user IDs on RDOS and DOS 1-9

N

- nesting limits
 - RDOS and DOS 1-7
 - RTOS 1-14
- NSYSGEN 1-1

O

- OL file 1-3, 1-8, 2-1
- operator commands on RDOS and DOS 3-1
- operator interrupts on RDOS and DOS 3-1
- overlay file 1-3, 1-8, 2-1
 - changing on AOS and AOS/VS 1-17
 - including features in, RDOS and DOS 1-5

P

- partitions, RDOS and DOS
 - primary 1-8
 - secondary 1-8
- PREDITOR 1-17
- prompt string 1-7
 - RDOS and DOS
 - RTOS 1-14

Q

- QTY 1-2

R

- RB file 1-3, 1-7
- real-time clock 1-2
- record locking 1-17
- recovery procedures on RDOS and DOS 2-2
 - auto restart 2-3
 - updating records 2-2
- release medium
 - loading onto AOS and AOS/VS 1-15
 - loading onto RDOS and DOS 1-2
- Resource Locking Server 1-17
- RTOSGEN, dialog 1-11

S

- save file, RDOS and DOS
 - for BASIC system 1-3, 1-7
- security considerations 3-10
- shutting down BASIC, RDOS and DOS 2-1
- single precision
 - AOS and AOS/VS 1-15, 1-17
 - RDOS and DOS 1-5
 - RTOS 1-13
- stacks, number of 1-1
- start-up procedures, RDOS and DOS 2-1
- string arithmetic
 - RDOS and DOS 1-5
 - RTOS 1-13
- subdirectories, RDOS and DOS 1-8
- subroutines
 - user-written on RDOS and DOS 1-5, 1-16
 - user-written on RTOS 1-13
- SV file 1-3, 1-7, 2-1
- SW file 1-3, 1-4, 1-7
- swapping, RDOS and DOS 1-4, 1-7, 1-3, 3-9
 - RDOS and DOS 3-9
 - to extended memory 3-9
- SYSGEN 1-1
- system calls
 - inactive 1-1
 - number active 1-1
- system manager programs 3-2
- system performance considerations, RDOS and DOS 3-9

T

- testing Extended BASIC, AOS and AOS/VS 1-18

U

- ULM 1-2
- unauthorized users on RDOS and DOS 3-11
- user core, maximum size of 1-7
- user identification file, RDOS and DOS 1-9

V

- virtual overlays, RDOS and DOS 1-6

reader comment form

Loading and Managing Extended BASIC

093-000119

Your comments will help us improve the quality of this publication. They will be carefully reviewed by the writers. Please refer to page numbers if appropriate.

DID YOU FIND THE MATERIAL:

- | | YES | NO | | YES | NO |
|-------------------|--------------------------|--------------------------|-----------------------|--------------------------|--------------------------|
| ● Useful? | <input type="checkbox"/> | <input type="checkbox"/> | ● Well illustrated? | <input type="checkbox"/> | <input type="checkbox"/> |
| ● Complete? | <input type="checkbox"/> | <input type="checkbox"/> | ● Well written? | <input type="checkbox"/> | <input type="checkbox"/> |
| ● Accurate? | <input type="checkbox"/> | <input type="checkbox"/> | ● Easy to read? | <input type="checkbox"/> | <input type="checkbox"/> |
| ● Well organized? | <input type="checkbox"/> | <input type="checkbox"/> | ● Easy to understand? | <input type="checkbox"/> | <input type="checkbox"/> |

COMMENTS:

HOW DID YOU USE THIS PUBLICATION?

- | | |
|---|---|
| <input type="checkbox"/> As an introduction to the subject | <input type="checkbox"/> As a student in a class |
| <input type="checkbox"/> For information about operating procedures | <input type="checkbox"/> As a reference manual |
| <input type="checkbox"/> To instruct in a class | <input type="checkbox"/> Other (<i>please explain</i>): |

Name _____

Title _____

Firm _____

Date _____

Street _____

State _____

City _____

Zip _____

First fold



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

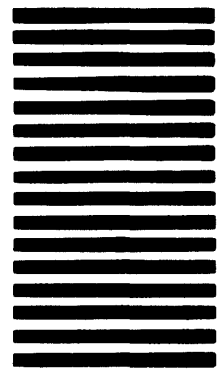
BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 26 WESTBORO, MASS 01580

POSTAGE WILL BE PAID BY ADDRESSEE:

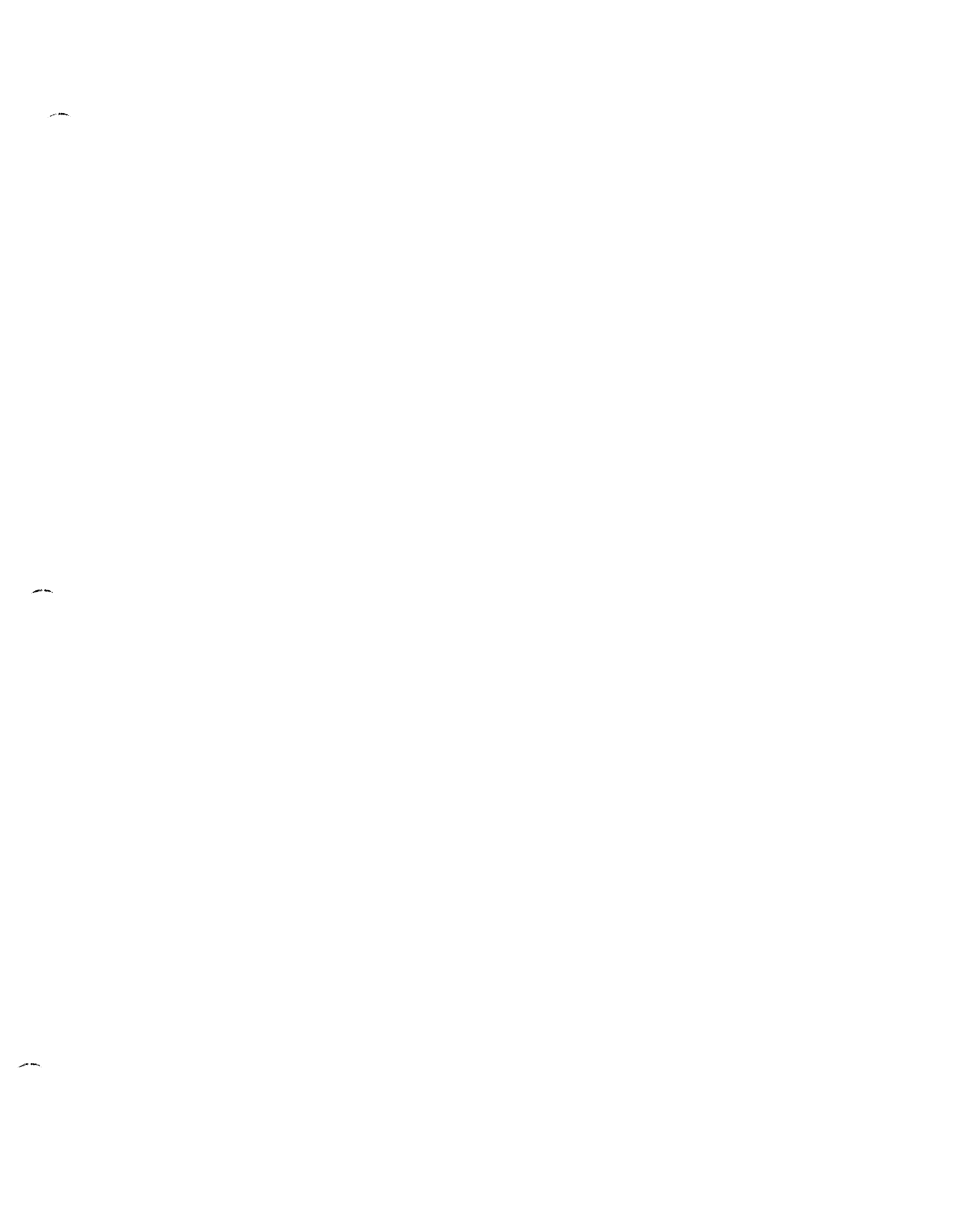


ATTN: SDD Documentation
62 Alexander Drive
Research Triangle Park, NC 27709
USA



CUT ALONG DOTTED LINE

Second fold



Data General Corporation, Westboro, MA 01580



093-000119-04