SERIES 60 (LEVEL 6)

GCOS 6 MOD 400 PROGRAMMER'S GUIDE ADDENDUM A

SUBJECT

Changes and Additions to the Manual

SPECIAL INSTRUCTIONS

Insert attached pages into the manual (Revision 0, dated January 1978) according to the collating instructions on the back of this cover. Except in the completely revised Section 5, change bars indicate new and changed information and asterisks denote deletions.

Note:

Insert this cover behind the manual cover to indicate that the manual is updated with this addendum.

SOFTWARE SUPPORTED

This update supports Release 0110 of the Series 60 (Level 6) GCOS 6 MOD 400 software system. See the Manual Directory of the *System Concepts* manual regarding later releases supported by this manual.

ORDER NUMBER

CB22A, Rev. 0

June 1978

20967 3678 Printed in U.S.A.

Honeywell

Collating Instructions

To update this manual, remove old pages and insert new pages as follows:

Remove	Insert
iii, blank	iii, blank
v, vi	v,vi
1-1, 1-2	1-1, 1-2
1-5, blank	1-5, blank
2-1, 2-2	2-1, 2-2
5-1 through 5-14	5-1 through 5-14
5-15, blank	5-15, blank
7-3 through 7-8	7-3 through 7-8

© 1978, Honeywell Information Systems Inc.

6/78 CB22A ٩,

r

MANUAL DIRECTORY

The following publications comprise the GCOS 6 manual set. The Manual Directory in the latest GCOS 6 MOD 400 Systems Concepts manual (Order No. CB20) lists the current revision number and addenda (if any) for each manual in the set.

Order	
No.	Manual Title
CB01	GCOS 6 Program Preparation
CB02	GCOS 6 Commands
CB03	GCOS 6 Communications Processing
CB04	GCOS 6 Sort/Merge
CB05	GCOS 6 Data File Organizations and Formats
CB06	GCOS 6 System Messages
CB07	GCOS 6 Assembly Language Reference
CB08	GCOS 6 System Service Macro Calls
CB09	GCOS 6 RPG Reference
CB10	GCOS 6 Intermediate COBOL Reference
CB20	GCOS 6 MOD 400 System Concepts
CB21	GCOS 6 MOD 400 Program Execution and Checkout
CB22	GCOS 6 MOD 400 Programmer's Guide
CB23	GCOS 6 MOD 400 System Building
CB24	GCOS 6 MOD 400 Operator's Guide
CB25	GCOS 6 MOD 400 FORTRAN Reference
CB26	GCOS 6 MOD 400 Entry-Level COBOL Reference
CB27	GCOS 6 MOD 400 Programmer's Pocket Guide
CB28	GCOS 6 MOD 400 Master Index
CB30	Remote Batch Facility User's Guide
CB31	Data Entry Facility User's Guide
CB32	Data Entry Facility Operator's Quick Reference Guide
CB33	Level 6/Level 6 File Transmission Facility User's Guide
CB34	Level 6/Level 62 File Transmission Facility User's Guide
CB35	Level 6/Level 64 (Native) File Transmission Facility User's Guide
CB36	Level 6/Level 66 File Transmission Facility User's Guide
CB37	Level 6/Series 200/2000 File Transmission Facility User's Guide
CB38	Level 6/BSC 2780/3780 File Transmission Facility User's Guide
CB39	Level 6/Level 64 (Emulator) File Transmission Facility User's
	Guide
CB40	IBM 2780/3780 Workstation Facility User's Guide
CB41	HASP Workstation Facility User's Guide
CB42	Level 66 Host Resident Facility User's Guide
CB43	Terminal Concentration Facility User's Guide

In addition, the following documents provide general hardware information:

Order

State of the second

No.	Manual Title
AS22	Honeywell Level 6 Minicomputer Handbook
AT04	Level 6 System and Peripherals Operation Manual
AT97	MLCP Programmer's Reference Manual
FQ41	Writable Control Store User's Guide



Contents

Section 1. Introduction

Guide to Using the Manual Set	1-1
Applications Programmer's Manual	
Guide	1-2
System Programmer's Manual Guide	1-2
Operator's Manual Guide	1-2
RBF and DEF User Manual Guide	1-5

Section 2. Operating Environments

Operator-Only Environment	2-1
All-Online Environment	2-1
Online/Batch Environment	2-1
Dedicated Application Environment	2-2
Mixed Environment	2-2

Section 3. User Terminal Startup

in the second

Startup with the Login Facility	3-1
Task Group-Specific Terminal Startup	3-1

Section 4. User Access to the System

Access by Loggin In	4-1	
Direct Login Terminal	4-1	
Abbreviated Login Terminal		
Full Login Terminal	4-1	
Command Processor as		
Lead Task	4-2	
Application as Lead Task	4-2	
Access through the Operator or		
Another User	4-2	
Serial Execution of Application		
Tasks	4-2	
Concurrent Execution of Application		
Tasks	4-3	
Concurrent Execution from Several	-	
Task Groups	4-3	
Execution of an Application from the		
Batch Task Group		
Execution from the Data Entry		
Facility (DEF)	4-4	
Access through the		
Operator Terminal	4-4	
	• •	
Section 5. Using the Editor		
Operator Terminal Typeout	5-6	
Explanation of Editor Actions	5-7	
•		
Section 6 Using the Assombler and		
Macro Proprocessor		
Samula Assemble I an many Cassier		
Sample Assembly Language Session	C 1	
(SMPMAC)		
Sample Assembly Language Multitask	~ ~	
Program (BKDUST)	0-0	

Section 7. Using the COBOL Compiler

Sample Card-to-Disk Program		
Volume and File Creation	7-2	
Source Loading	7-2	
Compiling with COBOL	7-2	
Linking	7-2	
Executing	7-3	
Sample COBOL Terminal Session		
(AC8111)	7-3	
Calling FORTRAN Routines from an		
Entry-Level COBOL Main Program	7-7	
Section 8. Using the FORTRAN		
Compiler		
Sample FORTRAN Terminal Session		
(MATINV)	8-1	
FORTRAN Chaining	8-1	

Section 9. Using the Sort

Figures

1-1	Applications Programmer Guide	
	to Manuals	1-3
1-2	System Programmer Guide to	
	Manuals	1-4
1-3	Operator Guide to Manuals	1-4
1-4	Guide for Using Manuals in a	
	Distributed Processing	
	Environment	1-5
5-1	Sample Editor Directives in File	
	SMPCMDFL	5-5
5-2	Terminal Responses from Sample	
	Editor Directives of Figure 5-1	5-6
5-3	Sample of Unexpanded Assembly	
	Language Program with Macro	
	Calls and Statements	
	(SMPMAC.P)	5-13
5-4	Sample of Unexpanded Macro	
	Routine (SAMPL1) Contained in	
	EXEC_LIB Directory	5-14
5-5	Sample of Unexpanded Macro	
	Routine (SAMPL2) Contained in	
	EXEC_LIB Directory	5-15
6-1	Sample Terminal Session	
	(SMPMAC)	6-1
6-2	Macro Preprocessor Output	• •
	(SMPMAC)	6-2
6-3	Cross Reference Listing	
	(SMPMAC)	6-3

6-4	Assembler Output Listing	
	(SMPMAC)	6-4
6-5	Linker Output Listing	
	(SMPMAC)	6-5
6-6	Sample Terminal Session	
	(BRDCST)	6-7
7-1	Sample Terminal Session	
	(AC8111)	7-3
7-2	Sample Listings for AC8111	7-4
7-3	COBOL Listing of COBFRT	7-8
7-4	FORTRAN Listing of FRTRAN	7-9
7-5	Operator Terminal Session for	
	COBFRT	7-9
8-1	Sample Terminal Session	
	(MATINV)	8-1
8-2	Source and Linker Output	
	Listing (MATINV)	8-2
8-3	Assembly Listing of Program	
	CHĂIN	8-7

FORTRAN Programs Calling	
the CHAIN Function	8-7
Linker Output for Chained	
Programs	8-9
Linker Directives for Chained	
Programs	8-15
Execution Output from Chained	
Programs	8-15
Sample Sort Terminal Session	9-1
Sample Sort Terminal Session	9-1
	FORTRAN Programs Calling the CHAIN Function Linker Output for Chained Programs Linker Directives for Chained Programs Execution Output from Chained Programs Sample Sort Terminal Session

Tables

÷

5-1 Explanation of Editor Actions 5-8

6/78 CB22A Ĭ

Section 1 Introduction

The GCOS 6 Mod 400 operating system for the Level 6 minicomputers provides a comprehensive set of system services which form a base for executing user-written applications, Honeywell-supplied applications, and program development tools. It provides an online, interruptdriven operation for multiple users and a single, low-priority batch operation typically used for program development and associated activities.

A number of different operating environments are possible, controlled in part by options exercised at system configuration, and in part by options chosen by the system operator at startup or at various times during the operating day. These environments are more fully described in Section 2, "Operating Environments."

Access to the system by users can be achieved in a variety of ways, again depending in part on system configuration options selected. These options are concerned mainly with the definition of local and/or remote terminal devices and how they are connected to the system. These are described in Section 3, "User Terminal Startup." Other access options, normally under the control of the system operator, are concerned with the procedures by which a user identifies himself (logs in) to the system through a connected terminal. This subject is treated in Section 4. "User Access to the System."

The remaining sections comprise descriptions and examples of the use of various system components: the Editor (Section 5), the Assembler and Macro Preprocessor (Section 6), the COBOL Compiler (Section 7), the FORTRAN Compiler (Section 8), and the Sort component (Section 9). Each of these sections presents terminal and/or line printer listings representing the actions performed. In these listings, heading lines may vary in detail depending on the component that initiated the listing or, in some cases, may be omitted. However, in actual use, the user will see heading lines consisting of three major fields of information, as shown below.

1. System Identification: GCOS6 MOD400- $\begin{cases} S \\ L \end{cases}$ rrr-mm/dd/hhmm

$$L - LA$$

- Release number of the operating system rrr

mm/dd/hhmm — Date/time when operating system was created (month, day, hour, and minute)

- 2. Component Identification: xxxxx-rrrr-mm/dd/hhmm
 - xxxxx Component name
 - rrrr Revision number of component

mm/dd/hhmm — Date/time that specified revision of component was created (month, day, hour, and minute)

3. Time of program execution: yyyy/mm/dd hhmm:ss.t

Date/time of program execution (year, month, day, hour, minute, second, and tenth of second)

GUIDE TO USING THE MANUAL SET

A guide to the use of the manual set is provided below. Information is tailored for specific classes of users — applications programmers, systems programmers, and operators. (As used in this guide, the applications programmer writes applications programs; the system programmer configures the system and defines the environment for each application; the operator operates the system from the operator terminal.) Included as a separate subsection is a guide for those who will use the Level 6 in a distributed processing environment.

APPLICATIONS PROGRAMMER'S MANUAL GUIDE

Figure 1-1 illustrates the suggested sequence in using the manuals. If you wish to start using the system by writing an application program, begin by using the *Programmer's Guide* manual. It illustrates: (1) various ways to gain access to the system, (2) a sample Editor session, and (3) for application languages, the procedure for performing program preparation and execution. Working with the small subset of commands used in the examples is a good approach to learning the system command set. This approach for getting started assumes that a system programmer has already configured and started up a suitable application environment. While using the system, you may wish to familiarize yourself with the system facilities described in the *System Concepts* manual.

Through examples, the *Programmer's Guide* illustrates how to use the system facilities. Other manuals provide reference material. The *Program Preparation* manual contains Editor directives (statements) to create and update an application language source unit. For each of the languages the appropriate language reference manuals contain the description of the language statements. Operating system dependencies, if any, that affect how you write the application are described in the *Programmer's Guide*. If the application uses communications, refer to the *Communications Processing* manual. Read the *Data File Organizations and Formats* manual if you require a better understanding of a language-supported file organization that is to be used in an application, or if you must calculate the size of a data file. You can use Monitor macro calls, as described in the *System Service Macro Calls* manual, in assembly language programs. Before your program can be entered for execution, it must be linked as described in the *Program*.

For program compilation or assembly and execution, the procedures described in the *Programmer's Guide* might be sufficient. To obtain more control over the execution of your program or utilize the system facilities more completely or efficiently, use the commands described in the *Commands* manual. If you wish to use the operator terminal, read the *Operator's Guide*. In many cases, the description of commands must be supplemented by system concepts described in the *System Concepts* manual. Rather than read all the conceptual material at one time, you may find it more meaningful to refer to it in conjunction with the appropriate reference material. The *Commands* manual also describes the utilities. An assembly language program, the Patch, Debug, and Dump utilities are described in the *Program Execution and Checkout* manual; file transmission from Level 6 to a host system is described in the *File Transmission* manual appropriate to the host system. Error messages and return status codes are listed in the *System Messages* manual.

SYSTEM PROGRAMMER'S MANUAL GUIDE

Figure 1-2 illustrates the suggested sequence for using the manuals. The System Building manual provides you with the configuration directives (statements) and startup procedures to configure and start up a MOD 400, a Remote Batch Facility (RBF), or a Data Entry Facility (DEF) system. You must know the conceptual material in the System Concepts manual in order to successfully use the configuration directives. To tailor an applications environment suitable for the intended application, use the operator commands described in the Operator's Guide manual. Error messages are listed in the System Messages manual. If you are working with an application that runs under the BES operating system, the System Concepts manual contains MOD 400 and BES compatibility considerations.

OPERATOR'S MANUAL GUIDE

Figure 1-3 illustrates the suggested sequence for using the manuals. Specific operator job functions must be determined by each installation; a large system might have a person assigned as an operator; a small system might have each programmer also act as an operator. The *Operator's Guide* indicates the system procedures performed through the operator terminal and describes operator commands used in system operation.

The Programmer's Guide contains examples using commands (described in the Commands manual) that are similar to operator commands. The System Concepts manual provides an understanding of the operating system. Note that the Operator's Guide describes using the

INTRODUCTION

CB22

operator terminal for operator functions to enter operator commands to the system task group, or for user functions to enter commands to a user task group. To run the utilities, use the commands (described in the *Commands* manual) entered through the operator terminal functioning as a user terminal. Error messages are listed in the *System Messages* manual.

GUIDE FOR USING THE MANUALS IN A DISTRIBUTED PROCESSING ENVIRONMENT

GCOS 6 Mod 400 supports the use of Level 6 in a distributed processing environment. Using Honeywell-supplied software packages, processing capability can be assigned to sites remote to the host computer system. With the functional links provided by Honeywell, a Level 6 can be configured as a host processor and specialized processing (i.e., forms data entry) assigned to remote terminals. Also, the user can develop links with non-Level 6 host processors and distribute the total processing load between the host and Level 6.

The software packages available to the user include the Data Entry Facility, Remote Batch Facility, Terminal Concentration Facility, File Transmission Facility, Host Resident Facility (Level 66), and IBM workstation emulation software. Figure 1-4 indicates the documentation available for the operation and use of such software. Configuration information, if applicable, is contained in the *System Building* manual.





驚嘆



Section 2

Operating Environments

The Mod 400 operating system allows a wide variety of operating environments, ranging from a single operator-controlled configuration to one in which the operator, other users, or a combination if both can control the configuration at any time during the operating day. This range of operating environments is described in this section.

OPERATOR-ONLY ENVIRONMENT

This environment is one in which a designated operator and a limited number of users (typically programmers developing application programs) use the system on a first-come firstserved basis for developing and testing programs. All work is done through the operator terminal, through either the system task group or a single online task group created by the system startup procedure. Certain functions can be performed through either of the two task groups; others can be done only through the system task group or the online task group — refer to the *Operator's Guide* and the *Commands* manuals for details on which functions can be performed from each task group.

ALL-ONLINE ENVIRONMENT

An all-online environment is one in which one or more users can concurrently use the facilities of the operating system to perform interactive tasks of any kind permitted by the command language described in the *Commands* manual, plus any user applications that can be invoked through the command processor. This latter category consists of user programs in the form of bound units that are called from a task group in which the command processor is declared as the lead task when the task group is created. A task group can also be created by the operator or another online user, declaring the application bound unit as the lead task; in this case the creation of the task group and its activation results directly in the execution of the declared bound unit, without the need to enter its name as a command.

An example of this kind of environment is one in which several task groups have the command processor as lead task and one or more other task groups have specific application programs such as the Data Entry Facility and user-created programs as lead tasks. The former task groups can be used for editing source program files, entering requests for jobs to be run in the batch task group (see below), requesting printouts of files, etc. Concurrent with these activities can be the execution of the user application programs constituting the latter set of task groups. From the user's point of view, each task group has the appearance of having control of the system.

ONLINE/BATCH ENVIRONMENT

This environment differs from the all-online environment only in that, in addition to the creation of the online task groups, a batch task group has also been created by the designated operator from the operator terminal. Once this task group has been created, any online task group having the command processor as its lead task can enter requests for jobs to be run through the use of the EBR (ENTER BATCH REQUEST) command. Typical of such batch jobs would be requests for compilations, links, application program checkout runs, and the like.

Creation and utilization of the batch task group requires the existence of at least the designated operator terminal, through which the batch task group is created and through which requests to it can be entered. Jobs run in the batch task group are normally controlled by a previously created file containing commands directing the execution of the jobs, and not by interactive dialog from a terminal. Section 4 contains additional information on the use of the batch task group.

DEDICATED APPLICATION ENVIRONMENT

This is an environment in which system startup or operator action subsequent to startup results in the creation of one or more task groups in which a user application, and not the command processor, is the lead task. In such an environment no interactive processing using system commands takes place; rather, whatever processing occurs is dependent on the nature of the application — e.g., data entry, an inventory application, etc.

MIXED ENVIRONMENT

The Mod 400 system does not restrict the user to any one of the foregoing environments at any given time. Given a large enough system, any of these can be combined with any others to provide concurrent interactive, batch, and dedicated operations on a selected terminal basis. That is, a selected set of terminals can be associated with interactive tasks, while others can be related to the dedicated application tasks.

Section 5 Using the Editor

Before studying this section, you should have a knowledge of the Editor operation. The Editor is described in the Program Preparation manual.

This section shows how the Editor is used to modify the contents of files, merge files and place macro routines in the macro library directory. The four files to be altered are SMPM01 (Example 1), SMPM02 (Example 2), SMPM03 (Example 3), and SMPM04 (Example 4). The directives that control the Editor are contained in a file ^ SYSMAC>SMPCMDIFL, Figure 5-1. This file comprises 56 lines, some of which include multiple directives.

This session of the Editor accomplishes the following functions:

- SMPM01 and SMPM02 are altered and written to files SMMPL1 and SMMPL2.
- SMMPL1 and SMMPL2 are combined to form file SMPMAC.P containing macro statements and calls to be processed by the macro preprocessor.
- SMPM03 and SMPM04 are altered and written as files SMMPL3 and SMMPL4 respectively.
- SMMPL3 and SMMPL4 are altered and written as macro library routine files SAMPL1 and SAMPL2, respectively, into the MACRO>EXEC_LIB directory.

Example 1:

漢律

File SMPM01 before Editing

1		TITLE	SMPMAC, 3/1/77' EDITOR/MACHU EXAMPLE
5	* INSERT I	LN 2 LIDM S	STATEMNT BEFORE THIS LNTHEN DEL THIS LN
3	SMPLM	MAC	P1=0, P2=2, P3='SAMPLE', P4='PR(GRAM', P5=ZERO, P6=(;
4	P7=),P6=T	NU, P9=5COMM	, PA=A, PH=H, PC=T2, PU=SAMPLE, PE=PROG2
5	* SET LUC	AL VALUES M	ITHIN MACRU ROUTINE *
6	LE	USE CHANGE	FUNCTION TO ADD SETA VALUE FOR THIS LN
7	L4	USE CHANGE	FUNCTION TO ADD SETA VALUE FOR THIS LN
8	15	USE CHANGE	FUNCTION TO ADD SETA VALUE FOR THIS LN
9	L6	USE CHANGE	FUNCTION TO ADD SETA VALUE FOR THIS LN
10	* ADU L7 3	SETA VALUE	W/CHANGE FUNCTION THEN DELT THIS LN
11	L8	SETA	XLOC
12	L9	SETA	XVAL
13	LA	SETA	[2'01']
14	LB	SETA	COMM
15	LZ	SETA	1, 1
16	*USED FOI'	T READ FUNC	T TO ADD "SMPMO2" PORTION TO FILE+

Example 2:

File SMPM02 before Editing

```
1
   * THESE UNPROTECTED COMMENT LINES WILL BE DRUPPED
5
   * WHEN MACRO PREPROCESSED.
3
4
    *
                                 (G1 INITIAL VALUE=S)
5
   764
             #L4
                       ?G1
                       ?IX(#LE,?PE)?G8?P1
6
              #15
7
    265
              #L6
                       ?P3?VL(35)?P4#LZ#LA
8
             #L7
                       ?G4
9
             #L7
                       ?P6?P8?GH?AL(?PC)?P7
10
                      ?SS(?P4,7,1)
             #L8
             #L9
                       ?VP(11)
11
   266
             #LB
12
                       ?67
                       ?P9+?G3
   ?G A
             ≠∟4
13
14
             ENDM
15 G3
             SETN
                       1
                       ZER0'
                                 (APOSTRUPHE'S DRUPPED WHEN SUBSTI.)
16
   G 4
              SETA
   65
             SETA
                       'NAME'
17
                        'SCOMM'
18
   66
             SETA
19
   G 7
             SETM
                       100
20
   GΑ
                        CON1!
             SETA
                       1,1
21
   GB
              SETA
55
   *
   **** THE FULLOWING PORTION OF CODE IS ADDED FROM "SMPLM" ****
23
24
   *
                                 (CALL IN-LINE MACRO ROUTINE)
25
              SHPLM,
56
   **** THE FULLOWING PORTION OF CODE IS ADDED FRUM "SAMPL1" ****
27
82
59
   CALL1
             SAMPL1
                       1,,,,,,+,150,;
   .. START. SC
30
31
   **** THE FULLOWING PORTION OF CODE IS ADDED FROM "SAMPL2" ****
32
53
             SAMPL2
34
   CALLS
                       SF,,,,,,,,;
35
   END
                      SMPMAC, START
36
```

5-2

.

Example 3:

A NEW

File SMPM03 before Editing

```
SAMPL1
               MAC
                          P1=0, P2=2, P3='SAMPLE', P4='PROGRAM', P5=ZER0, P6=(, P7=);
 1
    P8=Tw0, P9=SCOMM, PA=A, P8=B, PD=SAMPLE, PE=PROGRAM
 5
 3
 4
 5
    * SET LOCAL VALUES WITHIN MACRO ROUTINE *
 6
 7
    *
 8
    L4
               SETB
                          URG
    L5
 9
               SETB
                          DC
10
    L6
               SETH
                          LDR
11
   L7
               SETH
                          STR
12
   6 يا
               SETH
                          CALL
13
    ٢9
               SETB
                          LB
14
   LA
               SETR
                          681
15
   LB
               SETB
                          SLD
16
    LC
               SETH
                          1 = 1
17
                          [[2:32]]
    LD
               SETB
18
   LE
               SETB
                          PROG2.START2[,]NAME!
19
    *
50
    *
21
   * SET GLUBAL VALUES WITHIN MACKO ROUTINE *
55
   *
23
    *
24
    GΗ
               SETA
                          'ORG INTO COMMON'
25
   GG
               SETA
                          'ORG INTO INTERNAL LOC'
56
    GC
               SETA
                          'EXTERN VAL REFERENCE'
27
    GD
               SETA
                          COMMON REFERENCE!
28
    GE
               SETA
                          'EXTERNAL LOCATION REFERENCE'
               SETA
                          FORWARDS TEMP LAHEL REFERENCE!
29
    GF
30
    *
31
    * UNPROTECTED LINES OMITTED WHEN PRE-PROCESSED
35
    *
33
               ?L4
                          :49
                                                ?GH
34
                          ?VR(?P3,?PD)?G8?SR(?P4,?PE)
               ?L5
35
               ?L4
                          ?G4?P7?P8
                                                266
                          SR1, ?LC?PH
    ?PC
36
               ?L6
                                                           ?GC
37
               ?L7
                          5R1,<?GA
                                                ?GU
38
    [*]
    ?PD
39
               ?L6
                          5R1, <?PA
                                                ?GE
40
    [*]
41
               ?L8
                          PRUG2. ?SS (?LE, 7, 6) ?GHNAME
42
               ?L9
                          ?G4?P7?P1?G8?LC?VL(13)
43
               ?LA
                          >?P7$F
                                                ?GF
               ?LB
44
                          $$1?GB?LCZ'?CH(1,=2)?CH(2,=2)?CH(3,=2)?CH(4,=2)'
45
    ENDCL1
               ENDM
```



Example 4:

File SMPM04 before Editing

```
P1=0,P2=2,P3='SAMPLE',P4='PR0GRAM',P5=ZER0,P6=(,P7=);
    SAMPL2
             NAC
 1
    P8=TWU, P9=SCOMM, PA=A, P8=B
 2
   * SET LOCAL VALUES FITHIN MACRU ROUTINE *
L4 SETA >= (Z'1300')
 3
 4
    L4
                         >=[Z'1300']
               SETA
 5
   LA
                         10L0
               SETA
 6
    LÐ
                         SK1
 7
    LE
               SETA
                         PROG2.START [,] NAME!
    LG
               SETA
                         506
 8
 9
                         -32768
    LC
               SETIN
               DC "DELETE LINE ENDING IN S'S
10
    UEL TS
               SETN
                         32767
    LP
11
               SETIN
12
    Lü
                         0
13
               SETA
                         HEZ
    LI
14
    LY
               SETA
                         HLT
15
    LZ
               SETA
                         1,1
    * SET GLUBAL VALUES WITHIN MACKU RUNTINE *
16
               SETN
17
    G 7
                         -32765
18
               SETA
                          THACKWARDS TEMP LABEL REFERENCE!
    6 2
19
    65
               SETA
                         CTHL
50
    *
    * UNPROTECTED LINES OMITTED WHEN PRE-PROCESSED
21
55
    *
    2P1
                         ?P5?LZ?L4,=?LD
23
               ?LA
    ?P1
               ?LA
                         ?P5?L7?L4,=?L0
24
25
               ?LG
                          ?LD,?VG(3)
                         ?0,-$0
59
               115
                                    362
27
               ?L.Y
28
               ?65
                         ?PZ ?SS(?LE,1,5)
    OEL.
               DC "DELETE LINE DEGIDNING IN "
29
30
               IFINE
                         2G7, 2LP, GTEND
               FAIL
31
32
   GTEND
               GUTO
                         ENDIT
               DC POELETE LINE REFURE QUIT!
33
   DLET
34
    ENDORS
               ENDM
```



•

```
1
   R ^SYSMAC>SMPMO1
2
   X
   6,9CLE
                        ' PROG2.START2E, JNAME'
З
              SETA
 4
              SETA
                        EQU
   L4
5
   L5
              SETA
                        RESV
              SETA
                        TEXT
   L6
 67
   L7
              SETA
                        XDEF
8
    ! F
                        'EXEC_LIB', SAMPL1, SAMPL2!F!?
 9
   21
              LIBM
   =.-1;$K(SMMPL1)
10
   X
11
12
    1,$DX
13 R SMPMO2
   1,13V!P/#L/.-12;13G!P/#L/.-9;13S'#L'?L'P=
14
15
   1,$M(SMMPL2)X
16 R SMPMD3
17
   8,17S/SETB/SETA/8,17P
18
   1,$M(SMMPL3)X
19 R ASYSMAC>SMPM04
20 X29A
              IFE
                        ?G7,?LC,IFE1
21
              FAIL
                        ?P2,?LC,*
22 ENDIT
              IFNL
23 IFE1
              NULL ! F
   /SS..LE/!P/$$/!PD/^^/!PD
24
25
   1,$K(SMMPL4)
26 X1,$D
27
   B(SMMPL1)
2.8
   W ASYSMAC>SMMPL1.IN.A
29
   1,$DX
30 B(SMMPL2)
31
   W ASYSMAC>SMMPL2.IN.A
32
   1.95D
33
   B(SMMPL3)
34
   W ASYSMAC>SMMPL3
35
   1,$D
   B(SMMPL4)
36
   W ASYSMAC>SMMPL4
37
38
    i,$DX
   R SMMPL1.IN.A
39
40 /INSERT/!PD/ADD L7/!PD
   15R SMMPL2.IN.A
44
   X52!PD
42
43 E FO >SPD>LPTOO
   1,$!PW SMPMAC.P
44
45
   1,$D
   R SMMPL3
46
   X/L4/;/LE/S/SETB/SETA/
47
    1,$!PW ASYSRES>LDD>MACRO>EXEC_LIB>SAMPL1
48
49
   1,$D
50 R SMMPL4
51
    /DLET/D
52
   Q
53
    1,$!PW ASYSRES>LDD>MACRO>EXEC_LIB>SAMPL2
54
    X
   E FO
55
56
    G
```

Figure 5-1. Sample Editor Directives in File SMPCMDIFL

.

and the

OPERATOR TERMINAL TYPEOUT

The typeout produced at the operator's terminal during the editing process is shown in Figure 5-2. In this figure, the editor directive line that produced each line of the typeout is indicated by the circled numbers at the left of the typeout lines. Note that not every directive generates a typeout; e.g., the R (read) and W (write) directives. The lines of typeout that are produced in the editing of each of the four input files are indicated by brackets in the left margin.

In the type out, the response to the Editor directives begins after the line (H)EDIT-0100-11/21/0827.

EL GROUPSD GDJON-LINE DEBUG REV. 1976/11/20 1115 04 SYSREV. 4014 C 15H: HON GD 7 SYSMAC GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: LMD GHJRDY: CSHDIT-0100-11/21/0827 SMPM01 2 (SH) 16 -> (U) "SYSMAC>SMPCMDFL (SH) 18 -> MOD (U) "SYSMAC>SMPM01 GHJRDY: CSHDIT-0100-11/21/0827 SMPM01 2 (SH) 16 -> (U) "SYSMAC>SMPM01 GHJRDY: GHJRD											
GDJON-LINE DEBUG REV. 1976/11/20 1115 04 SYSREV. 4014 C 15H: RNN GHJRDY: UD GHJ SYSMAC GHJRDY: DD -LINE_LN 75 -IN "SYSMAC>SMPCMDFL BD -LINE_LN 75 -IN "SYSMAC>SMPCMDFL GHJ EDIT-0100-11/21/0827 (GHJ 16 -> (U) "SYSMAC>SMPM01 GHJ 2 GHJ 16 -> (U) "SYSMAC>SMPM01 12 - GHJ 18 -> MOD (0) "SYSMAC>SMPM01 12 - GHJ 18 -> MOD (0) "SYSMAC>SMPM01 13 - GHJ 18 -> MOD (0) "SYSMAC>SMPM01 14 - GHJ 18 -> (U) "SYSMAC>SMPM01 GHJ 1 - GHJ 18 -> MOD (0) "SYSMAC>SMPM01 14 - GHJ 18 -> (GJ) "SYSMAC>SMPM01 GHJ 1 - GHJ 18 -> MOD (0) "SYSMAC>SMPM01 GHJ 2 - GHJ 18 -> MOD (0) "SYSMAC>SMPM01 GHJ 3 -> (GJ) "SYSMAC>SMPM01 GHJ 3 -> (GJ) "SYSMAC>SMPM01 GHJ 4 - GHJ 18 -> MOD (0) "SYSMAC>SMPM01 GHJ 5 - 7G4 #L4 ?G1 (G1 INITIAL VALUE=S) GHJ 5 7G4 #L4 ?G1 (G1 INITIAL VALUE=S) GHJ 5 7G4 #L4 ?G1 (G1 INITIAL VALUE=S) GHJ 6 -> (J) "SYSMAC>SMPM02 GHJ 10 #L8 ?SS(PA47/1) GHJ 10 #L8 ?SS(PA47/1) GHJ 12 ?GGA #L4 ?P9+7G3 H4 - GHJ 18 (SMMPL1) GHJ 13 ?GGA #L4 ?P9+?G3 SMPM03 SMPM03 SMPM03 SMPM03 SMPM04 17 - GHJL5 SETA DC GHJ 6 SETA CALL GHJ 18 (SMMPL2) GHJ 8 -> (D) "SYSMAC>SMPM03 GHJ 18 (SMMPL2) GHJ 8 -> (D) "SYSMAC>SMPM03 GHJ 18 (SMMPL1) 18 - GHJL6 SETA BBT GHJL6 SETA SLD GHJ 8 -> (D) "SYSMAC>SMPM03 GHJ 18 (SMMPL1) 18 - GHJL6 SETA BBT GHJL6 SETA SLD GHJ 8 -> (D) "SYSMAC>SMPM03 GHJ 18 (SMMPL1) 18 - GHJL6 SETA SLD GHJ 4 - GHJ 8 -> (D) "SYSMAC>SMPM03 GHJ 18 (SMMPL1) 18 - GHJL6 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL7 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL7 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL7 SETA SLD GHJL7 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL7 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL6 SETA SLD GHJL7 SETA SLD GHJL7 SETA SLD GHJL6 SETA SLD GHJL7 SETA SLD G											
C 15H; FILE OLD CHART DIFFERENCE OF STERIOUS OF STERAC GH) RDY; OD 7 SYSMAC GH) RDY; DD GH > COD 7 SYSMAC GH > DIT_OLOO-11/21/0827 ED -LINE_LN 75 -IN "SYSMAC>SMPCMDFL (SH) BDIT_OLOO-11/21/0827 GH > DDIT OLOO-11/21/0827 GH > SYSMAC GH > DDIT OLOO-11/21/0827 (SH > DDIT OLOO SYSMAC>SMPM01 12 (SH > DIT OLOO SYSMAC>SMPM01 14 (SH > MOD (O) "SYSMAC>SMPM01 14 (SH > 2 THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH > 1 & CSMPPL) (GH > 2 THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH > 1 & SHAR > MOD (O) TYSMAC>SMPM01 (GH > 2 THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH > 1 & SHAR > MOD (O) TYSMAC>SMPM01 (GH > 2 THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH > 1 & SHAR > MOD (O) TYSYMAC>SMPM02 (GH > 2 THESE UNPROTECTED COMMENT LINE			CSD)ON-	LINI	E DEBUG	REV. 197	6/11/20	1115	0.4	SYSRE	N. 4014
RDN GKH RDY: UD GKH RDY: UD GKH RDY: UD GKH RDY: D ED SMPM01 GKH RDY: D SMPM01 GKH RDY: D D: SMPM01 GKH RDY: D: GKH RDY: D: SMPM01 GKH RDY: D: GKH RDY: D: GKH RDY: D: GKH RDY: CKH RDY: GKH RDY: CKH RDY: GKH RD: GKH RD: GKH RD: GKH RD: GKH RD: GKH RD:			C :SH:						04	5.5.	
SMPM02 GHD RDY: GHD RDY: GHD SYSMAC GHDRDY: ED -LINE_LN 75 -IN "SYSMAC>SMPCMDFL (SH) EDDT-0100-11/21/0827 (GHD EDDT-0100-11/21/0827 (GHD EDDT MODE 10 GHD EDDT MODE 10 GHD EDT MODE 10 GHD 18 -> MOD (0) "SYSMAC>SMPM01 11-GHD 18 -> MOD (0) "SYSMAC>SMPM01 12-GHD 0 -> (0) "SYSMAC>SMPM01 12-GHD 18 (SMMPL1) GHD 18 (SMMPL1) GHD 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED GHD 3 * WHEN MACRO PREPROCESSED. (GHD 4 * (GHD 4 * (GHD 4 * (GHD 5 7G4 #L4 ?G1 (G1 INITIAL VALUE=S) (GHD 6 #L5 ?11X(#LE,?PE)?GB?P1 (GHD 7 7G5 #L6 ?PE?VL(35)?P4#L2#LA (GHD 6 #L5 ?1X(#L2,?PE)?GB?P1 (GHD 7 7G5 #L6 ?PE?VL(35)?P4#L2#LA (GHD 6 #L5 ?1X(#L2,?PE)?GB?P1 (GHD 7 7G5 #L6 ?PE?VL(35)?P4#L2#LA (GHD 6 #L5 ?1X(#L2,?PE)?GB?P1 (GHD 10 #L8 ?S(?PA?T.1) (GHD 10 #L8 ?S(?PA?T.1) (GHD 10 #L8 ?S(?PA?T.1) (GHD 10 * (0) "SYSMAC>SMPM02 (GHD 13 ?GA #L4 ?P9+?G3 14-GHD 0 -> (0) "SYSMAC>SMPM02 (GHD 13 SETA CALL (GHD 6 SETA LB (GHD 6 SETA LB (GHD 6 SETA CALL (GHD 6 SETA CALL (GHD 8 SETA CALL (GHD 45 (SMMPL1) (GHD 8 -> (0) "SYSMAC>SMPM03 (GHD 8 -> (0) "SYSMAC>SMPM03 (GHD 8 -> (0) "SYSMAC>SMPM03 (GHD 18 (SMMPL1) (GHD 45 (SMMPL2) (GHD 45 (SMMPL2) (GHD 45 (SMMPL3)			RDN								
GWD -SYSMAC GHAPDY: LWD GHAPDY: LWD GHAPDY: D0 -LIMELLN 75 -IN "SYSMAC>SMPCMDFL SMPM01 PD -LIMELLN 75 -IN "SYSMAC>SMPM01 GHAPDIT-0100-11/21/0827 CGHD IS -> (0) "SYSMAC>SMPM01 GHAPDIT MODE 10 GHAPDIT MODE 11-GHAPDIT GGHAPDIT (GHAPDIT MODE 12-GHAPDIT MODE 13 GMAPDIT 14-GHAPDIT GHAPDIT 14-GHAPDIT GHAPDIT 12-GHAPDIT GHAPDIT 14-GHAPDIT GHAPDIT 14-GHAPDIT GHAPDIT 15 GHAPDIT 16 GHAPDIT 17 GHAPDIT 18 19 19 19 10 114 115 116 117 118 118 119 111			(SH)RDY	1	_						
SMPMO1 GH/SYSMAC GH/SYSMAC GH/SP/I ED -LINE_LN 75 - IN "SYSMAC>SMPCMDFL ED -LINE_LN 75 - IN "SYSMAC>SMPMD1 GH/SEDIT-0100-11/21/0827 2 GH/SEDIT-0100-11/21/0827 2 GH/SEDIT-0100-11/21/0827 2 GH/SEDIT-0100-11/21/0827 2 GH/SEDIT-0100-11/21/0827 2 GH/SEDIT-0100-11/21/0827 10 GH/SEDIT-0100-11/21/0827 11 GH/SEDIT-0100-11/21/0827 12 GH/SEDIT-0100-11/21/0827 12 GH/SEDIT-0100-11/21/0827 12 GH/SEDIT-0100-11/21/0827 12 GH/SEDIT-0100-11/21/0827 12 GH/SEDIT-0100-11/21/0827 12 GH/SEDIT-0100-11/21/0827 12 GH/SEDIT-0100-11/21/0827 12 GH/SEDIT-0100 GH/SEDIT-0100-11/21/0827 GH/SEDIT-0100 GH/SEDIT-0100-11/21/0827 GH/SEDIT-0100 GH/SEDIT-01			CWD SY	SMA(;						
SMPMO2 GH) 'SYSMAC GH) RDY: ED -LINE_LN 75 - IN 'SYSMAC>SMPCMDFL (SH) EDIT-0100-11/21/0827 2 (SH) EDIT-0100-11/21/0827 9 (SH) EDIT-0100-11/21/0827 9 (SH) EDIT-0100-11/21/0827 9 (SH) EDIT-0100-11/21/0827 9 (SH) EDIT-0100-11/21/0827 10 (SH) 2 (O) 'SYSMAC>SMPM01 11 (SH) 18 -> (0) 'SYSMAC>SMPM01 12 (SH) 18 (SMMPL1) 12 (SH) 18 (SMMPL1) 14 (SH) 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH) 3 * WHEN MACRO PREPROCESSED. (SH) 4 * (SH) 5 7G4 #L4 7G1 (G1 INITIAL VALUE=S) (SH) 4 * (SH) 5 7G4 #L4 7G1 (G1 INITIAL VALUE=S) (SH) 7 7G5 #L6 7P37VL(35)7P4#L2#LA (SH) 8 #L7 7G4 (SH) 9 #L7 7P67P87GB7AL(7PC)7P7 (SH) 10 #L8 7SS(7P4.7.1) (SH) 12 7G6 #L8 7G7 (SH) 13 7GA #L4 7P9+7G3 14 (SH) 14 (D -> (D) 'SYSMAC>SMPM02 (SH) 36 (SMMPL1) (SH) 15 SETA DC (SH) 18 (SMMPL1) (SH)L5 SETA LDR (SH)L5 SETA LDR (SH)L6 SETA LDR (SH)L6 SETA LDR (SH)L8 SETA SLD (SH)L8 SETA SLD (SH)L9 SETA LB 17 (SH)L8 SETA SLD (SH)L8 SETA SLD (SH)L9 SETA LB 17 (SH)L8 SETA SLD (SH)L8 SETA SLD (SH)L8 SETA SLD (SH)L9 SETA LB 17 (SH)L8 SETA SLD (SH)L8 SETA SLD (SH)L9 SETA LB 18 (SMMPL1) (SH)L8 SETA SLD (SH)L9 SETA LB 18 (SMMPL1) (SH)L8 SETA SLD (SH)L8 SETA SLD (SH)L9 SETA LB 18 (SMMPL1) (SH)L8 SETA SLD (SH)L9 SETA LB 18 (SMMPL1) (SH)L8 SETA SLD (SH)L9 SETA SLD (SH)L8 SETA SLD (SH)L9 SETA SLD (SH)L9 SETA SLD (SH)L8 SETA SLD (SH)L9 SET			ועאנאנא								
GH/RDY1 DD -LIME_LN 75 - IN "SYSMAC> SMPCNDFL (SH) EDIT-0100-11/21/0827 2 (SH) EDIT-0100-11/21/0827 3 (SH) EDIT-0100-11/21/0827 2 (SH) EDIT-0100-11/21/0827 2 (SH) EDIT-0100-11/21/0827 3 (SH) EDIT-0100-11/21/0827 3 (SH) EDIT-0100-11/21/0827 10 (SH) 2 11-(SH) 18 -> MOD (0) "SYSMAC>SMPM01 (SH) 18 (SMMPL1) 12-(SH) 18 (SMMPL1) 14 (SH) 1 (SH) 1 (SMPN01 (SH) 1 (SMPN01 (SH) 1 (SMPN01 (SH) 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH) 3 * WHEN MACRO PREPROCESSED. (SH) 4 * UA (SH) 5 7G4 (SH) 7 7G5 (SH) 7 7G5 (SH) 7 7G6 (SH) 8 #L7 (SH) 10 #L7 (SH) 12 7G6 (SH) 12 ?G6 (SH) 13 ?G7 (SH) 14 SETA (SH) 15			(SH)^SY	SMA	:						
ED -LINE_LN 75 -IN *SYSMAC>SMPCMDFL (SH) EDIT-0100-11/21/0827 2 (SH) EDIT MODE 9 GH:DEDIT MODE 10 (SH) 2 11 (GH) 2 (SH) 16 -> (0) *SYSMAC>SMPM01 11 (GH) 2 11 (GH) 18 12 (GH) 0 (GH) 1 18 (SH) 2 *THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (GH) 3 *WHEN MACRO PREPROCESSED. (GH) 4 *G (GH) 5 7G4 (GH) 6 /L17 (GH) 7 7G5 (GH) 7 7G6 (GH) 7 7G6 (GH) 7 7G6 (GH) 8 ?DS(P4/7)1 (GH) 12 7G6 (GH) 12 7G6 (GH) 13 7G6 (GH) 14 (D) * (GH) 13 7G6 (GH) 14 7G7 (GH) 15 SETA (GH) 16 (D) * (GH) 17 ?P6?P8?GB?AL(?PC)?P7 (GH) 18 (SMMPL1)			(SH)RDY	1	•						
SMPM01 (SH) EDT-0100-11/21/0827 9 GH) EDT MODE 10 GH) EDT MODE 10 GH) EDT MODE 11 GH) I 6 -> (O) "SYSMAC>SMPM01 11 GH) I 8 11 GH) I 6 -> (O) "SYSMAC>SMPM01 12 GH) I 8 11 GH) I 6 -> (O) "SYSMAC>SMPM01 12 GH) I 7 14 GH) I 7 15 GH) I 7 16 JU 7 17 GH 7 18 JU 7 19 JU 7 10 JU 7 11 GH 7 12 GG 4 13 7GA 4 14 GH 7 13 GA 4			ED -LIN	E_L	J 75 - IN	SYSMAC>S	MPCMDFL				
SMPM01 2 GRD 16 -> (0) "SYSMAC>SMPM01 10 GRD 2 11-GRD 18 -> MOD (0) "SYSMAC>SMPM01 11-GRD 18 (SMPL1) 18 (SMPL1) 12-GRD 0 -> (0) "SYSMAC>SMPM01 13-GRD 0 -> (0) "SYSMAC>SMPM01 14-GRD 18 (SMPL1) 14-GRD 7 (0) "SYSMAC>SMPM01 (GRD 2 "THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (GRD 3 * WHEN MACRO PREPROCESSED. (GRD 4 "GRD (GRD 7 G5 #L4 ?GI (GRD 7 G5 #L6 ?P37VL(3S)?P4#L2#LA (GRD 7 G5 #L6 ?P37VL(3S)?P4#L2#LA (GRD 9 #L7 ?P6?P87GB7AL(?PC)?P7 (GRD 11 #L9 ?VP(11) (GRD 12 ?G6 #LB (GRD 14 (SRDPL1) (SRDPL1) (GRD 0 -> (0) "SYSMAC>SMPM02 (GRD 14 (SRDPL2) (GRD SETA </th <th>1</th> <th></th> <th>(SH)ED</th> <th>IT-C</th> <th>100-11,</th> <th>/21/0827</th> <th></th> <th></th> <th></th> <th></th> <th></th>	1		(SH)ED	IT-C	100-11,	/21/0827					
SMPMO1 9 GH)2DIT MODE SMPMO1 10 GH) 2 11- 11- 12- GH) 18 -> MOD (0) "SYSMAC>SMPM01 12- GH) 0 -> (0) "SYSMAC>SMPM01 12- GH) 1 + (GH) 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (GH) 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (GH) 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (GH) 3 * WHEN MACRO PREPROCESSED- (GH) 4 * (GH) 5 7G4 #L4 7G1 (GI INITIAL VALUE-S) (GH) 6 #L5 71X(#L2.7P2)7G3P1 (GH) 7 7G5 #L6 7P37UL(35)7P4#L24LA (GH) 8 #L7 7G4 (GH) 8 #L7 7G4 (GH) 10 #L8 ?SS(?P4.7.1) (GH) 10 #L8 ?SS(?P4.7.1) (GH) 12 7G6 #L4 7P9+7G3 14- (GH) 12 7G6 #L4 7P9+7G3 14- (GH) 14 (GH) 14 (GH) 14 15- (GH) 14 15- (GH) 14 16- (GH) 15 SETA DC (GH)L7 SETA STR (GH)L8 SETA LDR (GH)L8 SETA CALL (GH)L8 SETA CALL (GH)L8 SETA SLD (GH)L8 SETA SLD (GH)L8 SETA SLD (GH)L8 SETA SLD (GH)L0 SETA SETA SETA SETA SLD (GH)L0 SETA SETA SETA SETA SETA SETA SETA SETA		2	(SH)	16	->	(0) °SYSMA	C> SMPM0 1				
SMMMUT 11 - (GH) 18 -> MOD (0) "SYSMAC>SMPM01 11 - (GH) 18 (SMMPL1) 12 - (GH) 18 (SMMPL1) 12 - (GH) 18 (SMMPL1) 14 - (GH) 1 * (GH) 1 * (GH) 1 * (GH) 1 * (GH) 3 * WHEN MACRO PREPROCESSED. (GH) 4 * (GH) 6 #L5 ?IX(#LE,?PE)?GB?P1 (GH) 6 #L5 ?IX(#LE,?PE)?GB?P1 (GH) 6 #L5 ?IX(#LE,?PE)?GB?P1 (GH) 7 ?G5 #L6 ?P3?UL(35)?P4#LZ#LA (GH) 9 #L7 ?P6?P8?GB?AL(?PC)?P7 (GH) 10 #L8 ?SS(?P4?1) (GH) 12 ?G6 #LB ?G7 (GH) 12 ?G6 #L4 ?P9+?G3 14 - (GH) 12 ?G6 #L4 ?P9+?G3 14 - (GH) 18 (SMMPL1) (GH) 18 (SMMPL1) (GH) 18 (SMMPL1) (GH)L6 SETA DC (GH)L6 SETA LDR (GH)L6 SETA LDR (GH)L6 SETA LDR (GH)L6 SETA LDR (GH)L9 SETA SIR (GH)L0 SETA SIR (GH)L1 (SMMPL1) (GH)L1 SETA SIR (GH)L0 SETA SIR (GH)L0 SETA SIR (GH)L1 (SMMPL1) (GH)L3 SETA SIR (GH)L1 (SMMPL3) (GH)L0 SETA SIR (GH)L2 (SMMPL3) (GH)L0 SETA SIR (GH)L1 (SMMPL3) (GH)L0 SETA SIR (GH)L1 (SMMPL3) (GH)L0 SETA SIR (GH)L2 (SMMPL3) (GH)L0 SETA SIR (GH)L3 SETA SIR (GH)L3 SETA SIR (GH)L3 SETA SIR (GH)L3 SETA SIR (GH)L3 SETA SIR (GH)L1 (SMMPL3) (GH)L0 SETA SIR (GH)L3	01401404	9	(SH)EDI	TM	DDE						
11- GHD 18 CSMPHD1 12- GHD 0 -> (1) "SYSMAC>SMPM01 (GH) 18 (SMPL1) (GH) 18 (SMPL1) (GH) 1 * (GH) 1 * (GH) 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (GH) 3 * WHEN MACRO PREPROCESSED. (GH) 4 * (GH) 5 7G4 #L4 ?GI (GI INITIAL VALUE=5) (GH) 5 7G4 #L4 ?GI (GI INITIAL VALUE=5) (GH) 6 #L5 ?IX(#LE.?PE)?GB?P1 (GH) (GH) 7 ?G5 #L6 ?P3?UL(35)?P4#L2#LA (GH) 9 #L7 ?P6?P8?GB?AL(?PC)?P7 (GH) 11 #L8 ?SS(?P4.7.1) (GH) 12 ?G6 #LB ?G7 (GH) 13 ?GA #L4 ?P9+?P33 14- (GH) 14 SETA DC (GH) 18 (GMPL2) SETA	SIVIPIVIUT		(SH) 7(EH)	2	-> MOD	(D) TOYOMA	C CHDM01				
$SMPM02 = \begin{cases} GHD & 0 \rightarrow & (D)^{*}SYSMAC>SMPM01 \\ 12 & GHD & 1 & 8 \\ (GHD & 1 & 4 \\ (GHD & 2 & * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED \\ (GHD & 3 & * WHEN MACRO PREPROCESSED. \\ (GHD & 5 & 7G4 & 4L4 & ?G1 & (G1 INITIAL VALUE=S) \\ (GHD & 5 & ?G4 & 4L4 & ?G1 & (G1 INITIAL VALUE=S) \\ (GHD & 6 & 4L5 & ?1X(4LE, ?PE?)?GB?P1 \\ (GHD & 6 & 4L5 & ?1X(4LE, ?PE?)?GB?P1 \\ (GHD & 8 & 4L7 & ?G4 \\ (GHD & 8 & 4L7 & ?G4 \\ (GHD & 8 & 4L7 & ?G4 \\ (GHD & 10 & 4L8 & ?SS(7P4.7.1) \\ (GHD & 12 & ?G6 & 4L8 & ?G7 \\ (GHD & 12 & ?G6 & 4L8 & ?G7 \\ (GHD & 13 & 7G6 & 4L4 & ?P9*7G3 \\ 14 & GHD & 14 \\ (GHD & 14 & GHD \\ 15 & (GHD & 14 \\ (GHD & 14 & GHD \\ 15 & (GHD & 16 & (SMMPL1) \\ (GHD & 16 & SETA & DC \\ (GHD & 16 & SETA & DC \\ (GHD & 17 & SETA & STR \\ (GHDL8 & SETA & LDR \\ (GHDL4 & SETA & CALL \\ (GHDL5 & SETA & LB \\ (GHDL6 & SETA & LB \\ (GHDL6 & SETA & LB \\ (GHDL7 & SETA & SETA & SLD \\ (GHDL8 & SETA & SETA & GRG \\ (GHDL6 & SETA & CALL \\ (GHDL6 & SETA & CALL \\ (GHDL7 & SETA & SETA & SETA \\ (GHDL8 & SETA & CALL \\ (GHDL8 & SETA & SETA \\ (GHDL8 & SETA \\ (GH$		11-	(SH)	18	⇒r muD	(SMMPL1)	O-SHPMUI				
SMPM02 12 (SH) 18 (SMMPL1) 14 (SH) 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH) 3 * WHEN MACRO PREPROCESSED. (SH) 4 * (SH) 4 * (SH) 5 7G4 #L4 ?GI (GI INITIAL VALUE=S) (SH) 6 #L5 ?IX(#LE.?PE)?GB?PI (SH) (SH) 6 #L7 ?G4 ?GI (GI INITIAL VALUE=S) (SH) 6 #L7 ?G4 ?GI (GI INITIAL VALUE=S) (SH) 7 ?G5 #L6 ?P37VL(35)?P4#L2#LA (SH) 8 #L7 ?G4 ?P2 (SH) 9 #L7 ?P6?P8?GE?AL(?PC)?P7 (SH) 10 #L8 ?SSYP4/2#L2#LA (SH) 12 ?G6 #LB ?G7 (SH) 13 ?GA #L4 ?P9+?G3 14 (SH) 14 SETA DC (SH) 16 (SHMPL1) SETA SETA (SH)			-(s H)	ŏ	->	(0) SYSMA	C> SMPM0 1				
SMPM02 (SH) 1 14 (SH) 2 * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH) 4 * (SH) 4 * (SH) 4 * (SH) 5 ?G4 ?G1 (G1 INITIAL VALUE=S) (SH) 6 ?L3 ?IX(?LE, ?PE)?GB?P1 (SH) 7 ?G5 ?L6 ?P3?VL(35)?P4?L2?LA (SH) 8 ?L7 ?G4 ?G1 (SH) 9 ?L7 ?P6?P8?GB?AL(?PC)?P7 (SH) 10 ?L8 ?SS(?P4.7.1) (SH) 11 ?L9 ?VP(11) (SH) 12 ?G6 ?LB ?G7 (SH) 13 ?GA ?L4 ?P9+?G3 14 (SH) 14 ?STSMAC>SMPM02 ?SH (SH) 16 (SMMPL1) ?SH ?SMPM02 (SH) 16 SETA DC ?SH ?SH (SH) 16 SETA LB ?SH ?SH (SH) 16		12-	(\$H)	18		(SMMPL1)					
$SMPM02 = \begin{cases} 14 & (SH) & 2 & * THESE UNPROTECTED COMMENT LINES WILL BE DROPPED (SH) & 3 & * WHEN MACRO PREPROCESSED. (SH) & 5 & 7G4 & #L4 & 7G1 & (G1 INITIAL VALUE=S) (SH) & 6 & #L5 & 71X(#LE, ?PE)?GB?P1 (SH) & 6 & #L5 & 71X(#LE, ?PE)?GB?AL(2#LA (SH) & 8 & #L7 & ?G4 (SH) & 8 & #L7 & ?G4 (SH) & 8 & #L7 & ?G4 (SH) & 9 & #L7 & ?PE?GB?AL(?PC)?P7 (SH) & 10 & #L8 & ?SS(?PA,7,1) (SH) & 11 & #L9 & ?VP(11) (SH) & 12 & ?G6 & #LB & ?G7 (SH) & 13 & ?GA & #L4 & ?P9+?G3 (SH) & 14 & SETA & DC (SH) & 18 & (SMMPL1) (SH) & 18 & (SMMPL2) (SH)L4 & SETA & DC (SH)L6 & SETA & LDR (SH)L7 & SETA & STR (SH)L8 & SETA & CL (SH)L8 & SETA & CL (SH)L8 & SETA & LB (SH)L9 & SETA & LB (SH)L8 & SETA & SETA & SLD (SH)L8 & SETA & SLD (SH)L8 & SETA & SLD (SH)L0 & SETA & SMPM03 (SH)L0 & SETA & (Z*32*] (SH)L8 & SETA & SMPM03 (SH)L9 & SETA & SMPM03 (SH)L9 & SETA & (SMMPL3) (SH)L8 & SETA & SMPM03 (SH)L9 & SETA & (SMMPL1) (SH)L8 & SETA & SMPM03 (SH)L9 & SETA & SMPM03 (SH)L18 & (SMMPL1) (SH)L4 & SETA & SMPM03 (SH)L9 & SETA & (SMMPL1) (SH) 18 & (SMMPL1) (SH) 18 & (SMMPL1) (SH) 45 & (SMMPL3) (SH)L9 & SETA & SMPM03 (SH) 18 & (SMMPL3) (SH) 45 & (SH) $			-(SH)	1	+						
$SMPM02 = \begin{bmatrix} (3H) & 3 & * & WHEN MACRO PREPROCESSED. \\ (3H) & 4 & * \\ (3H) & 4 & * \\ (3H) & 5 & 7G4 & #L4 & 7G1 & (G1 INITIAL VALUE=5) \\ (3H) & 6 & #L5 & 7IX(#LE,7PE)7GB7P1 \\ (3H) & 7 & 7G5 & #L6 & 7P37VL(35)7P4#LZ#LA \\ (3H) & 7 & 7G5 & #L6 & 7P37VL(35)7P4#LZ#LA \\ (3H) & 8 & #L7 & 7P67P87GB7A1(?PC)?P7 \\ (3H) & 10 & #L8 & 7S5(7P4,7,1) \\ (3H) & 11 & #L9 & 7VP(11) \\ (3H) & 12 & 7G6 & #LB & 7G7 \\ (3H) & 13 & 7GA & #L4 & 7P9+7G3 \\ (3H) & 12 & 7G6 & #LB & 7G7 \\ (3H) & 13 & 7GA & #L4 & 7P9+7G3 \\ 14 & -(3H) & 14 & (3MMPL1) \\ (3H) & 16 & (SMMPL1) \\ (3H) & 16 & (SMMPL2) \\ (3H)L4 & SETA & DC \\ (3H)L5 & SETA & DC \\ (3H)L6 & SETA & LDR \\ (3H)L6 & SETA & LDR \\ (3H)L6 & SETA & LDR \\ (3H)L6 & SETA & LB \\ (3H)L6 & SETA & SETA \\ (3H)L6 & SETA & SETA \\ (3H)L8 & SETA & SETA \\ (3H)L8 & SETA & CALL \\ (3H)L9 & SETA & LB \\ (3H)L0 & SETA & SETA \\ (3H)L0 & SETA & SETA \\ (3H)L0 & SETA & SETA \\ (3H)L0 & SETA & (2*32*) \\ (3H) & 18 & (SMMPL1) \\ (3H) & 18 & (SMMPL3) \\ \end{bmatrix}$		14-	(SH)	2	* THESE	UNPROTECT	ED COMMEN	IT LINE	S WIL	L BE	DROPPED
SMPM02 (SH) 4 (SH) 5 7G4 (L4 7G1 (G1 INITIAL VALUE=S) (SH) 6 (L5 71X(#LE,?PE)?GB?P1 (GH) 7 7G5 #L6 7P37VL(35)?P4#L2#LA (GH) 8 #L7 7G4 (GH) 9 #L7 7P67P8?GB?AL(?PC)?P7 (GH) 10 #L8 ?SS(?P4,7,1) (GH) 12 7G6 #L9 ?VP(11) (GH) 12 7G6 #L4 ?P9+?G3 [GH) 1 [GH) 12 (GH) 14 (GH) 15 (GH)L4 SETA DC (GH)L5 SETA DC (GH)L5 SETA DC (GH)L5 SETA DC (GH)L6 SETA LDR (GH)L6 SETA LB (GH)L9 SETA LB (GH)L SETA LB (GH)L SETA SETA SETA LB (GH)L SETA LB (GH)L SETA SETA [GH)L [GH)L SETA [GH)L [GH]L [GH]L [GH]L [GH]L [GH]L [GH]L [GH]L [GH]L			(SH)	3	* WHEN	MACRO PREF	PROCESSED	•			
$SMPM02 = \begin{pmatrix} GH1 & GH1 &$			(8H)	4	* 2GA	#1 A	201			* * ^ *	
$SMPM02 = \begin{pmatrix} SH3 & 7 & 7G5 & HL6 & PB37VL(35)7P4HL2HA \\ SH3 & HL7 & 7G4 \\ SH3 & HL7 & 7G4 P4L2HA \\ SH3 & P537VL(35)7P4HL2HA \\ SH3 & SH3 & P537VL(35)7P4HL2HA \\ SH3 & SH3 & P537VL(35)7P4HL2HA \\ SH4 & SH3 & SH3 \\ SMPM03 & (SH3 & SH3 & SH3 \\ SMPM03 & (SH3 & SH3 & SH3 & SH3 \\ SMPM03 & (SH3 & SH3 & SH3 & SH3 & SH3 \\ SMPM03 & (SH3 & SH3 & SH3 & SH3 & SH3 & SH3 \\ SMPM03 & (SH3 & SH3 & SH3 & SH3 & SH3 & SH3 & SH3 \\ SMPM03 & (SH3 & SH3 & $			(SH)	6	104	#15	212(4)	.F. 20F)	20838	1 IAL	VALUE=37
SMPM02 (SH) 8 #L7 ?G4 (SH) 9 #L7 ?P6?P8?GB?AL(?PC)?P7 (SH) 10 #L8 ?SS(?P4,7,1) (SH) 11 #L9 ?VP(11) (SH) 12 ?G6 #L8 ?G7 (SH) 12 ?G6 #L8 ?G7 (SH) 12 ?G6 #L8 ?G7 (SH) 13 ?GA #L4 ?P9+?G3 14 (SH) 14 (SH) 8 15 (SH) 14 (SH) 8 (SH) 36 (SMMPL1) 36 (SMMPL2) (SH) 36 (SMMPL2) (SH) 36 (SMMPL3) SMPM03 (SH) SETA LB SETA LB (SM) SETA LB SETA SETA SETA (SM) SETA S			(SH)	7	?G5	#L6	7P37VI	.(35)?P	- 400 · F	LA	
SMIPMU2 SMIPMU2 = (SH) 9 + L7 ?P6?P8?GB?AL(?PC)?P7 (SH) 10 + L8 ?SS(?P4,7,1) (SH) 11 + L9 ?Q6 (SH) 12 ?Q6 + LB ?Q7 (SH) 13 ?GA + L4 ?P9+?G3 14 - (SH) 14 (SH) 14 (SH) 16 (SMMPL1) (SH) 18 (SMMPL1) (SH) 18 (SMMPL2) (SH)L6 SETA DC 17 - (SH)L6 SETA DC (SH)L7 SETA STR (SH)L8 SETA CALL (SH)L8 SETA CALL (SH)L8 SETA LB (SH)L9 SETA LB (SH)L9 SETA LB (SH)L9 SETA SLD (SH)L0 SETA '-' (SH)L0 SETA '-' (SH)L0 SETA '-' (SH)L0 SETA SMPM03 (SH) 18 (SMMPL1) (SH) 18 (SMMPL1) (SH) 18 (SMMPL2) (SH) 18 (SMMPL1) (SH) 18 (SMMPL2) (SH) 18 (SMMPL2) (SH) 18 (SMMPL2) (SH) 18 (SMMPL2) (SH) 18 (SMMPL1) (SH) 45 (SMMPL3)	01401400		(SH)	8		#17	?G4				
$SMPM03 = \begin{bmatrix} (SH) & 10 & #L8 & ?SS(?P4.7.1) \\ (SH) & 11 & #L9 & 7VP(11) \\ (SH) & 12 & 7G6 & #LB & ?G7 \\ (SH) & 13 & ?GA & #L4 & ?P9+?G3 \\ 14 - (SH) & 14 & \\ (SH) & 14 & \\ (SH) & 14 & \\ (SH) & 16 & (SMMPL1) \\ (SH) & 36 & (SMMPL2) \\ 15 - (SH) & 36 & (SMMPL2) \\ (SH)L4 & SETA & DC & \\ 17 - (SH)L6 & SETA & LDR & \\ (SH)L7 & SETA & STR & \\ (SH)L7 & SETA & STR & \\ (SH)L8 & SETA & CALL & \\ (SH)L8 & SETA & CALL & \\ (SH)L8 & SETA & CALL & \\ (SH)L8 & SETA & SLD & \\ (SH)L6 & SETA & SLD & \\ (SH)L0 & SETA & SLD & \\ (SH)L0 & SETA & SLD & \\ (SH)L0 & SETA & (Z*32*) & \\ 18 - (SH) & 18 & (SMMPL1) & \\ (SH) & 18 & (SMMPL1) & \\ (SH) & 36 & (SMMPL2) & \\ (SH) & 45 & (SMMPL3) & \\ \end{bmatrix}$	SIVIPIVIUZ	-	(SH)	9		#L7	?P6?P	B?GB?AL	(7PC)	?P7	
$SMPM03 = \begin{cases} (SH) & 11 & 4L9 & 7UP(11) \\ (SH) & 12 & 7G6 & 4LB & 7G7 \\ (SH) & 13 & 7GA & 4L4 & 7P9+7G3 \\ (SH) & 14 & ENDM \\ (SH) & 14 & \\ (SH) & 14 & \\ (SH) & 14 & \\ (SH) & 16 & \\ (SH)$			(SH)	10		#L8	? 55 (?)	24,7,1)			
$SMPM03 = \begin{cases} (3H) & 12 & 766 & #LB & 767 \\ (3H) & 13 & 76A & #L4 & 7P9+7G3 \\ (3H) & ENDM \\ 14 - (3H) & 14 \\ (3H) & 36 & (3MMPL1) \\ (3H) & 36 & (SMMPL2) \\ (3H) & 36 & (SMMPL3) \\ (3H) & 36 & (SMMPL1) \\ (3H) & 18 & (SMMPL1) \\ (3H) & 18 & (SMMPL3) \\ (3H) & 45 & (SMMPL3) \\ \end{cases}$			(SH)	11		#L9	?VP(1)				
$SMPM03 = \begin{bmatrix} (SH) & (SH$			(SH)	12	766	#LB	767				
$SMPM03 = \begin{bmatrix} 14 & -16 & $			C(SH)	13	FND	#1.4	rrytr	33			
$SMPM03 = \begin{bmatrix} (SH) & 0 & -> & (0) & SYSMAC>SMPM02 \\ 15 & (SH) & 18 & (SMMPL1) \\ (SH) & 36 & (SMMPL2) \\ (SH)L4 & SETA & ORG \\ (SH)L5 & SETA & DC \\ (SH)L6 & SETA & LDR \\ (SH)L7 & SETA & STR \\ (SH)L8 & SETA & CALL \\ (SH)L8 & SETA & CALL \\ (SH)L8 & SETA & LB \\ (SH)L9 & SETA & LB \\ (SH)L9 & SETA & LB \\ (SH)L8 & SETA & SLD \\ (SH)L9 & SETA & SLD \\ (SH)L0 & SETA & (Z*32*) \\ (SH)L0 & SETA & (Z*32*) \\ (SH) & 0 & -> & (0) & SYSMAC>SMPM03 \\ (SH) & 18 & (SMMPL1) \\ (SH) & 36 & (SMMPL2) \\ (SH) & 45 & (SMMPL3) \\ \end{bmatrix}$		14	(SH)	14		•					
$SMPM03 = \begin{bmatrix} 15 & (SH) & 18 & (SMMPL1) \\ (SH) & 36 & (SMMPL2) \\ (SH)L4 & SETA & ORG \\ (SH)L5 & SETA & DC \\ (SH)L6 & SETA & LDR \\ (SH)L7 & SETA & LB \\ (SH)L7 & SETA & CALL \\ (SH)L8 & SETA & CALL \\ (SH)L8 & SETA & CALL \\ (SH)L8 & SETA & LB \\ (SH)L9 & SETA & LB \\ (SH)L0 & SETA & SETA \\ (SH)L0 & SETA & SETA \\ (SH)L0 & SETA & (Z*32*) \\ (SH)L0 & SETA & (Z*32*) \\ (SH) & 18 & (SMMPL1) \\ (SH) & 36 & (SMMPL2) \\ (SH) & 45 & (SMMPL3) \\ \end{bmatrix}$			GH)	0	->	(0) "SYSMA	C> SMPM02				
SMPM03 Image: Constraint of the system o		15_	(SH)	18		(SMMPL1)					
SMPM03 $ \begin{array}{ccccccccccccccccccccccccccccccccccc$			L(SH)	36		(SMMPL2)					
SMPM03 17 (SH)L6 SETA LDR (SH)L7 SETA STR (SH)L7 SETA SETA (SH)L8 SETA CALL (SH)L9 SETA LB (SH)L0 SETA SETA (SH) 18 (SMMPL1) (SH) 36 (SMMPL3)			(SH)L4		SETA						
SMPM03 Image: Superior of the second sec		17	(SH)LS		551H 5574	קרו לל					
SMPM03 (SH)L8 SETA CALL (SH)L9 SETA LB (SH)L4 SETA BBT (SH)L5 SETA SETA (SH)L6 SETA SETA (SH)L0 SETA (Z*32*) (SH)L0 SETA (SMMPL1) (SH)L1 SETA (SMMPL2) (SH)45 (SMMPL3)			(SH)L7		SETA	STR					
SMPM03 (SH) L9 SETA LB (SH) LA SETA BBT (SH) LB SETA SLD (SH) LC SETA (C) (SH) LD SETA (C) (SH) LB SETA (C) (SH) LD SETA (C) (SH) LB SETA (C) (SH) LB SETA (C) (SH) A5 (SMMPL2) (SH) (SH) A5 (SMMPL3)			(SH)L8		SETA	CALL					
17- (SH)LA SETA BBT 17- (SH)LB SETA SLD (SH)LC SETA '=' (SH)LD SETA [Z'32'] (SH) 0 -> (0) "SYSMAC>SMPM03 (SH) 18 (SMMPL1) (SH) 36 (SMMPL2) (SH) 45 (SMMPL3)	SMPM03		_(SH) L9		SETA	LB					
17- (SH) LB SETA SLD (SH) LC SETA '=' (SH) LD SETA [Z'32'] (SH) B -> (0) "SYSMAC>SMPM03 (SH) 18 (SMMPL1) (SH) 36 (SMMPL2) (SH) 45 (SMMPL3)			(SH) LA		SETA	BBT					
(SH)LC SETA [Z*32*] (SH)LD SETA [Z*32*] (SH) 0 -> (0) "SYSMAC>SMPM03 18 (SH) 18 (SH) 18 (SMMPL1) (SH) 36 (SMMPL2) (SH) 45 (SMMPL3)		1/	(SH) LB		SETA	SLD					
(SH) 0 * SYSMAC>SMPM03 (SH) 18 (SMMPL1) (SH) 36 (SMMPL2) (SH) 45 (SMMPL3)			(SH)LC		SEIF	5789	1941				
18 (SH) 18 (SMMPL1) (SH) 36 (SMMPL2) (SH) 45 (SMMPL3)				8	->	(0) "SYSMA	C> SMPM0 3				
18 (SH) 36 (SMMPL2) (SH) 45 (SMMPL3)			(SH)	18		(SMMPL1)					
(\$H) 45 (SMMPL3)		18—	(SH)	36		(SMMPL2)					
			(SH)	45		(SMMPL3)					



USING THE EDITOR



	⊏(\$¥)	34	
	(\$H)	18	(SMMPL1)
20—	(SH)	36	(SMMPL2)
	(SH)	45	(SMMPL3)
	T(SH)	28	?G5 ?PZ ?SS(?LE,1,5)
24—	(SH)	10	DELTS DC 'DELETE LINE ENDING IN \$'S
	(SH)	28	DEL DC DELETE LINE BEGINNING IN
	CSH)	36	-> MOD (0) [°] SYSMAC>SMPM04
	(SH)	18	(SMMPL1)
26—	(SH)	36	(SMMPL2)
	(SH)	45	(SMMPL3)
	C(SH)	36	(SMMPL4)
	(SH)	0	(0) SYSMAC> SMPM04
20	(SH)	0	-> (SMMPLI) SYSMAC>SMMPLI
29-	(SH)	36	(SMMPL2)
	(SH)	45	
	C(CH)	30	
	(SH)	0	
20	(SH)	0	(SMMPL2) SYSMAC>SMMPL2
30	(SH)	ů	(SMMP1.3) *SYSMAC>SMMP1.3
	(SH)	0	-> (SMMPL4) SYSMAC>SMMPL4
	(SH)	้้้	* INSERT LN 2 LIBM STATEMNT BEFORE THIS LN. THEN DEL THIS LN
40-	(SH)	11	* ADD L7 SETA VALUE W/CHANGE FUNCTION THEN DELT THIS LN
	T(SH)	0	(0) SYSMAC>SMPM04
	(SH)	0	(SMMPLI) SYSMAC>SMMPL1
40	(SH)	0	(SMMPL2) SYSMAC>SMMPL2
42	(SH)	0	(SMMPL3) SYSMAC>SMMPL3
	(SH)	52	-> MOD (SMMPL4) "SYSMAC>SMMPL2
	(SH)	52	*USED EDIT READ FUNCT TO ADD "SMPM02" PORTION TO FILE*
	(SH)	0	(0) SYSMAC>SMPM04
	(SH)	0	(SMMPLI) SISMAC>SMMPLI
47-	(SH)	U	(SMMPL2) SISMAC>SMMPL2
	(SH)	U 8 E	
	(CH)	43 151511	TO DUFFFRS FXIST. QUIT DEFERRED
	(SH)M)))))))))))))))))))	$(0) ^{SYSMAC>SMPM0A}$
	(SH)	n n	(SMMPL1) SYSMAC>SMMPL1
F A	(SH)	ň	(SMMPL2) SYSMAC>SMMPL2
54	(SH)	Ő	(SMMPL3) SYSMAC>SMMPL3
	(SH)	35	-> (SMMPL4) ^{200B02>LDD>MACR0>EXEC_LIB>SAMPL2}
	GH)RI	DY	



EXPLANATION OF EDITOR ACTIONS

Table 5-1 describes the actions performed by the Editor as it processes the directive file. The entries under the heading Line No. designate lines in the directive file Figure 5-1. The entries under the heading Terminal Typeout are abbreviated versions of those in Figure 5-2. For lines with multiple directives, they show which directive caused the typeout.

The assembly language program with unexpanded macro calls created from Example 1 and Example 2 is shown in Figure 5-3. The directive that caused it to be printed is contained in line 44.

Figures 5-4 and 5-5 are the unexpanded macro routines created from Example 3 and Example 4 respectively. The directives that caused them to be printed are included in lines 48 and 53 respectively.

Line No.	Editor Directive Description	Terminal Typeout	
1	R ^ SYSMAC>SMPM01		
	Read the 16-line file (example 1) into the current buffer (0) .		
2	X Display the status of the current buffer (denoted by \rightarrow). Sixteen lines were read into current buffer (0).	16- >(0)	
3	6,9Ctext Change lines 6 through 9 of the buffer with this text for line 6.		
4	Text for line 7.		
5	Text for line 8.		
6	Text for line 9.		
7	Text, this additional line is inserted after the previous four lines were changed. Note that the Editor recognizes tab characters.		
8	!F		
	Terminate input mode and enter edit mode.		
9	2Itext!F!?		
	Insert text before line 2.		
	Terminate input mode.		
	Display current mode.	EDIT MODE	
10	=1;\$K(SMMPL1)	_	
	Display current line pointer.	2	
	Move the current line pointer back one line to a new current line pointer position. Copy the lines from that position to the last line in the current buffer into auxiliary buffer, SMMPL1.		
11	X		
	Display status of current and auxiliary buffers. There are 18 lines in current buffer (0), which has been modified (MOD) since it was read in, and 18 lines in auxiliary buffer SMMPL1.	18-> MOD (0) 18 (SMMPL1)	
12	1,\$DX		
	Delete the first through last line of the current buffer (0). Display status of buffers.	0->(0) 18 (SMMPL1)	
13	R SMPM02		
	Read 36-line file (example 2) into the current buffer (0).		
14	1,13V!P/#L/12;13G!P/#L/9;.S'#L'?L'!P=		
	For lines 1 through 13, display line numbers and all lines that do not contain the expression #L.	1* 2*THESE	
		· • •	
	Move the current line pointer back 12 lines from line 14 to line 2.	4 [.] 5?G4#L4	
	For lines 2 through 13, display all lines and their line numbers	•	
	containing the expression #L.		
		13?GA #L4	

TABLE 5-1. EXPLANATION OF EDITOR ACTIONS

5-8

6/78 CB22A ٠

•

Ĵ

J

Line No.	Editor Directive Description	Terminal Typeout
	Move the current line pointer back nine lines to line 5 and substitute ?L for $\#L$, in that line and the next 8 lines. Note that the apostrophe is used as a delimiter.	
	Print current line.	ENDM
	Print current line pointer value.	14
15	1,\$M(SMMPL2)X	
	Move line 1 through the last line of the current buffer to auxiliary buffer SMMPL2. The contents of the current buffer (0)	
	are erased. Display the status of the buffers.	0->(0) 18 (SMMPL1)
		36(SMMPL2)
16	R SMPM03	
	Read 45-line file (example 3) into current buffer (0).	
17	8,17S/SETB/SETA/8,17P	
	For lines 8 through 17, substitute SETA for SETB.	
	Print lines 8 through 17 without line numbers.	L4 SETA
		LD SETA
18	1.\$M(SMMPL3)X	
	Move line 1 through last line of the current buffer into auxiliary buffer SMMPL3 and erase buffer (0).	
	Display buffer status.	$0 > (0) \dots$
		36 (SMMPL2)
		45 (SMMPL3)
19	R ^ SYSMAC>SMPM04	
	Read the 34-line file (example 4) into the current buffer (0) .	
20	X29A	
	Display buffer status, 34 lines are currently in buffer (0).	34->(0) 18 (SMMPL1) 36 (SMMPL2) 45 (SMMPL3)
	After line 29, append four lines of text. Text for line 30.	
21	Text for line 31.	
22	Text for line 32.	
23	text!F	
20	Last line of text (line 33).	
	Terminate input mode and enter edit mode.	
24	/SSLE/!P/\$\$/!PD/ ^ ^ /!PD	
	Search the current buffer for the first occurrence of the	
	List the line and its line number.	28?G5 ?PZ ?SS(?LE

TABLE 5-1 (CONT). EXPLANATION OF EDITOR ACTIONS

USING THE EDITOR

,

ĺ

144

ł

6/78 CB22A

Line No.	Editor Directive Description	Terminal Typeout
	Locate a line that ends with \$ as the last character. List the line	
	and its number; then delete the line.	10DELT\$ DC'D\$'\$
	Locate a line beginning with $^{\wedge}$. List, then delete the line and its	
	line number.	28 ^ DEL DC `DELETE
25	1, \$K(SMMPL4)	
	Copy the current buffer contents from first through last line into auxiliary buffer SMMPL4.	
26	X1,\$D	
	Display the status of the buffers.	36->MOD(0) 18 (SMMPL1) 36 (SMMPL2) 45 (SMMPL3) 36 (SMMPL4)
	Delete first through last line of current buffer.	
27	B(SMMPL1) The auxiliary buffer, SMMPL1, is made the current buffer prior to writing.	
28	W ^ SYSMAC>SMMPL1	
	Write the current buffer contents as a file whose pathname is $^{\circ}$ SYSMAC>SMMPL1.	
29	1,\$DX Delete the first through last line of the current buffer. Display the buffer status. The pointer points to current buffer, SMMPL1.	0(0) 0->(SMMPL1) 36 (SMMPL2) 45 (SMMPL3) 36(SMMPL4)
30	B(SMMPL2)	
	The auxiliary buffer, SMMPL2, is made the current buffer prior to writing.	
31	W ^ SYSMAC>SMMPL2	
	Write the current buffer contents as a file whose pathname is ^ SYSMAC>SMMPL2.	
32	1,\$D	
	Delete the first through last line of the current buffer.	
33	B(SMMPL3)	
	The auxiliary buffer, SMMPL3, is made the current buffer prior to writing.	

USING THE EDITOR

٠

	TABLE 5-1 (CONT). EXPLANATION OF EDITOR ACTIONS							
Line No.	Editor Directive Description	Terminal Typeout						
34	W ^ SYSMAC>SMMPL3 Write the current buffer contents as a file whose pathname is ^ SYSMAC>SMMPL3.							
35	1,\$D Delete the first through last line of the current buffer.							
36	B(SMMPL4) The auxiliary buffer, SMMPL4, is made the current buffer prior to writing.							
37	W ^ SYSMAC>SMMPL4. Write the current buffer contents as a file whose pathname is ^ SYSMAC>SMMPL4.							
38	1,\$DX Delete the first through last line of the current buffer. Display the status of the buffers. SMMPL4 is the current buffer. All the buffers have been cleared.	0 (0) 0 (SMMPL1) 0 (SMMPL2) 0 (SMMPL3) 0->(SMMPL4)						
39	R SMMPL1							
	Read the file SMMPL1 into the current buffer, SMMPL4.							
40	/INSERT/!PD/ADD L7/!PD							
	Locate the first line containing the expression, INSERT, list it and its line number, and then delete it.	3* INSERT LN						
	Starting at the current line, locate the first line containing the expression, ADD L7, list it and its line number, and then delete it.	11* ADD L7 SETA						
41	15R SMMPL2							
	Read the file, SMMPL2, into the current buffer after line 15 of the buffer. Two files are being merged.							
42	X52!PD							
	Display the status of the buffers. Current buffer, SMMPL4, now has 52 lines.	0 (0)						
		52->MOD(SMMPL4)						
	List line 52 then delete it	52 *USED EDIT						
13	F = FO = SPD > LPT00							
10	The Execute directive allows you to execute the ECL command FO to change the output file from the operator's terminal to the line printer.							
44	1,\$PW SMPMAC.P List the first through last line of current buffer on the line printer (Figure 5-3). Write the current buffer as a file whose pathname is SMPMAC.P.							

*

*

(

after liter.

(



Line No.	Editor Directive Description	Terminal Typeout
45	1,\$D	
	Delete the first through last line of the current buffer.	
46	R SMMPL3	
	Read the file, SMMPL3, into the current buffer, SMMPL4.	
47	X/L4/;/LE/S/SETB/SETA/	
	Display the status of the buffers.	0 (0) 0 (SMMPL1) 0 (SMMPL2) 0 (SMMPL3) 45->(SMMPL4)
	Locate the first line containing the expression, L4. Starting with the line containing L4 through the line containing the expression LE, substitute SETA for all occurrences of SETB.	
48	1,\$!PW ^Z00B02>LDD>MACRO>EXEC_LIB>SAMPL1 List the first through last line of the current buffer on the line printer (Figure 5-4).	
	Write the current buffer as a library routine file whose path- name is ^Z00B02>LDD>MACRO>EXEC_LIB>SAMPL1.	
49	1,\$D	
	Delete the first through last line of the current buffer.	
50	R SMMPL4	
	Read the file, SMMPL4, into the current buffer.	
51	/DLET/D	
	Locate and delete the line containing the expression DLET.	
52	Q	
	Quit. The quit is deferred since a buffer has been modified and has not been written to a file. You have <i>one</i> more chance to write the contents of the current buffer as a file.	MODIFIED BUFFERS EXIST
53	1,\$!PW ^Z00B02>LDD>MACRO>EXEC_LIB>SAMPL2	
	List the first through last line of the current buffer on the line printer (Figure 5-5).	
	Write the current buffer contents as a library routine file whose pathname is 200B02 DD>MACRO>EXEC_LIB>SAMPL2.	
54	X	
	Display buffer status. Status is always displayed on the operator's terminal even though the output file is the printer.	0 (0) 0 (SMMPL1) 0 (SMMPL2) 0 (SMMPL3) 35->(SMMPL4)
55	E FO	
	The Execute directive allows you to execute the ECL command FO to change the output file from the line printer back to the operator's terminal.	
56	Q	
	Quit. Exit from the Editor.	

TABLE 5-1 (CONT). EXPLANATION OF EDITOR ACTIONS

USING THE EDITOR

5-12

6/78 CB22A ۲

٠

TITLE SMPMAC, '3/1/77' EDITOR/MACRU EXAMPLE 1 'EXECTLIB', SAMPL1, SAMPL2 2 LIRW P1=0, P2=2, P3='SAMPLE', P4='PROGRAM', P5=ZER0, P6=(; 3 SMPLM MAC 4 P7=),P8=TW0,P9=\$COMM,PA=A,PB=B,PC=T2,PD=SAMPLE,PE=PROG2 * SET LOCAL VALUES WITHIN MACRO ROUTINE * 5 SETA ' PROG2.START2[,]NAME' LE 6 7 L4 SETA FQU 8 L5 SETA RESV TEXT 9 SETA L6 10 L7 SETA XDEF SETA XLOC 11 LB L9 12 SETA XVAL SETA [2'01'] 13 LA SETA COMM 14 ĻВ 15 SETA 1, 1 LZ 16 17 * THESE UNPRUTECTED COMMENT LINES WILL BE DROPPED * WHEN MACRU PREPRUCESSED. 18 19 * 50 ?G4 ?L4 (G1 INITIAL VALUE=8) ?G1 ?IX(?LE,?PE)?G8?P1 215 21 55 ?65 ?L6 ?P3?VL(35)?P4?LZ?LA 23 ?17 ?G4 24 ?L7 ?P6?P8?G5?AL(?PC)?P7 25 ?L8 ?SS(?P4,7,1) ?VP(11) ?L9 56 27 ?G6 ?LB ?G7 ?P9+?G3 ?GA ?L4 85 29 ENUM SETIN 30 G 3 1 ZEROT (APUSTROPHE'S DROPPED WHEN SUBSTI.) 31 G4 SETA G5 SETA 'NAME' 32 SCOMM! 33 SETA Gб 34 G7 SETIN 100 35 GΑ SETA 'COM1' 1,1 36 GВ SETA 37 **** THE FOLLOWING PURTIUN OF CODE IS AUTED FROM "SMPLM" **** 38 39 * (CALL IN-LINE MACRO ROUTINE) SMPLM. 40 41 **** THE FULLOWING PURTION OF CODE IS ADDED FROM "SAMPL1" **** 42 43 CALLI SAMPL1 1,,,,,+,150,; 44 ,,START,SC 45 46 **** THE FULLOWING PORTIUN OF CODE IS ADDED FROM "SAMPL2" **** 47 48 SAMPL2 49 CALLS \$F,,,,,,,;LINK 50 SMPMAC, START 51 END

Figure 5-3. Sample of Unexpanded Assembly Language Program with Macro Calls and Statements (SMPMAC.P)

「日本の

SAMPL1 P1=0,P2=2,P3='SAMPLE',P4='PROGRAM',P5=ZER0,P6=(,P7=); MAC 1 P8=Tw0, P9=SCOMM, PA=A, P8=B, PD=SAMPLE, PE=PROGRAM S 3 * 4 * 5 SET LOCAL VALUES WITHIN MACRO ROUTINE * × 6 * 7 * 8 L4 SETA ORG 9 LS SETA DC 10 SETA LDR L6 L7 SETA STR 11 12 L8 SETA CALL 13 L9 SETA LB 14 LA SETA 8BT 15 LB SETA SLD 16 LC SETA 1=1 (Z'32') 17 LD SETA 18 LE SETA 'PROG2.START2[,]NAME' 19 × 20 * * SET GLOBAL VALUES WITHIN MACRO ROUTINE * 21 55 * 23 ٠ 24 GH SETA 'ORG INTU COMMON' 'ORG INTO INTERNAL LOC' 25 SETA GG 59 GC SETA 'EXTERN VAL REFERENCE' 27 GD SETA 'COMMON REFERENCE' 28 GΕ SETA 'EXTERNAL LOCATION REFERENCE' 29 GF SETA 'FORWARDS TEMP LABEL REFERENCE' 30 × 31 * UNPROTECTED LINES OMITTED WHEN PRE-PROCESSED 32 * 33 ?L4 ?P9 ?GH 34 ?L5 ?VR(?P3, ?PD)?GB?SK(?P4, ?PE) 35 ?L4 2G42P72P8 ?GG 36 ?PC SR1, ?LC?PB ?GC ?L6 \$R1,<?GA 37 ?L7 **?G**D 38 [*] ?PD 39 ?L6 \$R1,<?PA ?GE 40 [*] 41 ?L8 PROG2. ?SS (?LE, 7, 6) ?GBNAME ?G4?P7?P1?G8?LC?VL(13) ?L9 42 43 ?LA >?P7\$F ?GF \$\$1?GB?LCZ'?CH(1,=2)?CH(2,=2)?CH(3,=2)?CH(4,=2)' ?LB 44 45 ENDCL1 ENDM

> Figure 5-4. Sample of Unexpanded Macro Routine (SAMPL1) Contained in EXEC_LIB Directory

USING THE EDITOR

1	SAMPL2	MAC	P1=0,P2=2,P3='SAMPLE',P4='PRUGRAM',P5=ZER0,P6=(,P7=);
2	P8=TW0, P9:	=\$COMM,PA=	A,PB=8
3	* SET LOC.	AL VALUES I	NITHIN MACRO ROUTIDE *
4	L4	SETA	>=[2'1300']
5	LA	SETA	IOLD
6	LD	SETA	SR1
7	LE	SETA	PROG2.START[,]NAME'
8	LG	SETA	SÜK
9	LC	SETIN	-32768
10	LP	SETN	32767
11	Lü	SETN	0
12	LI	SETA	8E Z
13	LY	SETA	HLT
14	LZ	SETA	۲,۱
15	* SET GLO	BAL VALUES	WITHIN MACRU ROUTINE *
16	G 7	SETN	-32768
17	G 2	SETA	BACKWARDS TEMP LABEL REFERENCE!
18	65	SETA	CTRL
19	*		
50	* UNPROTE	CTEU LINES	OMITTED WHEN PRE-PROCESSED
21	*		
55	?P1	?LA	?P5?LZ?L4,=?LU
23	?P1	?LA	?P5?LZ?L4,=?LU
24		?LG	?LD,?VG(3)
25		?LI	3FD'-2C 3C5
56		?LY	
27		?G5	?PZ ?SS(?LE,1,5)
28		IFE	?G7,?LC,1FE1
29		FAIL	
30	ENDIT	IFNL	\$P2,3LC,*
31	IFE1	NULL	
32		IFNE	?G7,?LP,GTEND
33		FAIL	
34	GTEND	GOTO	ENUIT
35	ENDCL2	ENDM	



18 B



EXECUTING

The internal file names 0A and 0C translate to logical file numbers 01 and 03, respectively, and must be associated with the pathnames or the physical devices through a GET or ASSOC command. To execute the program, enter CARDIN.

Enter the following commands:

GET 01 >SPD>CDR00 GET 03 ^VOL03>FILES>OLD_MASTER CARDIN

SAMPLE COBOL TERMINAL SESSION (AC8111)

Figure 7-1 illustrates an operator terminal session in which a system is configured, and the COBOL program AC8111 is compiled, linked, and executed on that system. The Entry Level COBOL compiler is specified in this session. To secify the Intermediate COBOL compiler, change the COBOL command and the LINKER LIB directive as follows.

COBOLI AC8111 -LO -COUT >SPD>LPT00 LIB ^ZSYS51>ZCIRT;LINK AC8111;MAP;QT

The LINKER LIB directive directs the Linker to search the secondary directory for COBOL run-time routines required for linking. To execute the program, enter AC8111.

```
COBOLI ASTCOB1>SOURCE>ACC2D8>AC8111 -LO -XREF -COUT >SPD>LPTOD
(%H) COBOL1 0200 05/08/1050
($H) 0000 ERRORS
($H)END COMPILATION
($H)RDY:
LINKER AC8111 - COUT >SPD>LPTDD
($H)LINKER-0100-04/05/0704
LIB <ZCIRT
LINK ZOAC8111; MP;QT
($H)ROOT AC8111
($H)LINK DONE
($H)RDY:
AC8111
($H)Q208NUA011001
($H)0208NUA011001
                         1234
                         PPPP
($H)
($H)RDY:
```

Figure 7-1. Sample Terminal Session (AC8111)

Figure 7-2 is a listing of the program AC8111, its compiled object text, and the output from the Linker. The program was compiled using the entry-level compiler.

=GCOS6 MUD400-S110-04/26/0916 COBOLI 0200 -L0 XREF AC8111 78/05/08 SOURCE PROGRAM IDENTIFICATION DIVISION. *PROGRAM 0208401101.COBOL FROM 0208ACC.ARCHIVE. 5 PROGRAM-ID. AC8111. 3 4 ENVIRONMENT DIVISION. 5 CONFIGURATION SECTION. 6 SOURCE-COMPUTER. LEVEL-6. OBJECT-COMPUTER. LEVEL-6 PROGRAM COLLATING SEQUENCE IS ASCTI. 7 8 DATA DIVISION. 9 WORKING-STORAGE SECTION. 10 01 DDSPLYPER. 11 05 NDSPLYFIX. 12 10 FILLER PIC X(13) VALUE "0208NUA011001". 13 10 OTCASE PIC XX VALUE SPACES. VALUE SPACES. 14 10 FILLER PIC XX 15 10 OSTATUS PIC XX VALUE SPACES. 16 10 FILLFR PIC XX VALUE SPACES. 05 ODSPLYVBL. 17 18 VALUE SPACES. 10 DACTRESLT PIC X(12) 19 FILLER PIC XX 10 VALUE SPACES. 20 DEXPRESLT PIC X(12) 10 VALUE SPACES. 21 10 FILLFR PIC XX VALUE SPACES. 55 01 SUMMARYS. 23 05 SUM-LINE PIC X(7) VALUE "1 2 3 4". 24 05 RESULTS. 25 PIC XX. 10 TESTIR 26 TEST2R PIC XX. 10 27 10 TEST3R PIC XX. TEST4R 28 PIC XX. 10 29 * * TEST GO TO--FORWARD AND BACK * * * 30 PROCEDURE DIVISION. ANFANG. 31 32 DISPLAY ODSPLYFIX. 33 TO PARA-3. GO 34 WBA1. 35 MOVE "GO TO PARA-3" TO GEXPRESLT. "FELL THRU" TO DACTRESLT. 36 MOVE "01" TO QTCASE. "F" TO TESTIR. 37 MOVE 38 MOVE TESTIR. 39 DISPLAY NDSPLYREC. 40 PARA-1. 41 "P" TO MOVE TEST3R. 42 GO TO EOJ1. 43 WBA2. 44 MOVE "GO TO FOJ1" TO NEXPRESLT. "FELL THRU" TO GACTRESLT. 45 MOVE 46 MOVE "04" TO GTCASE. TO TESTAR. HFH 47 MOVE 48 DISPLAY ODSPLYREC. 49 PARA-2. "P" 50 MOVE TO TEST2R. 51 60 PARA-1. TO 52 WBA3. "GO TO PARA-1" TO DEXPRESLT. "Fell thru" <u>to dactreslt.</u> 53 MOVE -54... MOVE "03" TO OTCASE. "F" TO TEST3R. 55 MOVE 56 MOVE 57. DISPLAY ODSPLYREC. 58 PARA-3. 59 MOVE "P" TO TESTIR. 60 60 TO PARA-2. 61 WBA4. 62 MOVE "GO TO PARA-2" TO GEXPRESLT. MOVE "FELL THRU" TO QACTRESLT. 63

Figure 7-2. Sample Listings for AC8111



4	MOVE "02"	TO OTCAS	SE.		
5	MOVE "F"		28.		
6	DISPLAY NDS	PLYREL.			
-/ E		TO TEST	/I D		
0	MOVE SPA	CES TO	4κ. οτι	- ASE	
0	MOVE SPA	CES TO	QS1	TATUS.	
1	DISPLAY NDS	PLYFIX SU	M-1	INE.	
2	MOVE SPA	CFS TO	٩Q	SPLYFIX.	
3	DISPLAY ODS	PLYFIX	RFS	SULTS.	
4	STOP RUN.				
5 F	ND COPOL.				
=6C085 M0D4	00-5110-04/26/	0916 COBC	IL I	0200 -1.0	XREF AC8111 78/05/0
DATA ALLOCA	TION MAP				
LEVEL NO.	NAME	LHAD	AL	PICTURE	
IORKING-STOR	AGE SECTION	0000		×(0000/19) 1	
		0000		X(000021)	
10	FTLLER	0000		X(000013)	
	OTCASE	0006	н	X(000002)	
10	FILLER	0007	н	X(000005)	
10	QSTATUS	0008	н	X(000005)	
10	FILLER	0009	н	X(000005)	
05	QDSPLYVBL	0004	н	X(000028)	
10	GACTRESLT	0004	н	X(000012) \	
10	OFVDOFOLT	0010		X(000002)	(DATA ALLOCATION MAP)
10	FTILED	0017	н	X(000002)	· · · · · · · · · · · · · · · · · · ·
01	SUMMARYS	0019		X(000015)	
05	SUM-LINE	0019		X(000007)	
05	RESULTS	001C	н	X(000008)	
	TFSTIR	0010	н	X(000005)	
10	TFST2P	001D	н	X(000002)	1
10	TEST3R	001E	н	x(000002)	1
	- +EST4R		H	x1.000.002.	F
			7	\sim	
Ť	Ť	T		<u> </u>	(HALF-WORD INDICATOR.
					DESIGNATED BY H)
T		L			(STARTING ADDRESS OF DATA
		_			(DATA NAME)
"					(GROUP AND ELEMENIARY IIE
					(GROUP LEVEL NUMBERS)

Figure 7-2 (cont). Sample Listings for AC8111

.

I





USING THE COBOL COMPILER

7-6

6/78 CB22A

34 WBA1 NO REFERENCES WBA2 WBA3 43 NO REFERENCES 52 NO REFERENCES WBA4 NO REFERENCES 61 LINKFR-0100-11/23/1258 GC056 MOD400-S100-11/29/0620 LINKED ON: 1901/01/01 0002:44.1 -SAF BU= ACA111 AC8111 01/01/01 COBOL REV. 0200 DATE 01/01/01 TIME 0000 . ZCRTYU 770208 HRS ASSEMBLER 2.49 06/02/77 1340.3 EDT THU (C) COPYRIGHT 1976 BY HONFYWELL INFORMATION SYSTEMS INC ZCSTOP 770208 HRS ASSEMBLER 2.49 06/02/77 1336.9 EDT THU (C) COPYRIGHT 1976 BY HONFYWELL INFORMATION SYSTEMS INC ZCRTFR 770208 HRS ASSEMBLER 2.49 06/02/77 1934.4 EDT THU (C) COPYRIGHT 1976 BY HONEYWELL INFORMATION SYSTEMS INC ** AC8111 LINK MAP 1901/01/01 0002:44.1 0033 **START **LOW 0000 0381 **HIGH **CURRENT 0381 **EXT DEFS ZHCOMM P 0000 ρ ZHREL 0000 ** R00T 0000 AC8111 0000 AC8111 0033 7CMAIN 0031 ZCRIYU 0289 7CRTY1 02F1 7CRTY2 0312 ZCRTY3 0334 ZESTOP 033E ZCSTOP 033E ZCRTFR 0341 ZCRTER 0353 **UNDEF * AC8111 0000 * ZCRTYU 0289 * ZCSTOP 033E ZCRTER 0341 * ****** ROOT AC8111 ****** HIGHEST OVLY /NUM OF SYMS 1 ****** SAF ******* ROOT AC8111 BASE 0000 ST 0033 -...I HIGH=0381 ***** *SIZE OF ROOT AND STATIC OVLYS= 0381 HI REL RCD= 9 ******* LINK DONE ******

Figure 7-2 (cont). Sample Listings for AC8111

重要

CALLING FORTRAN ROUTINES FROM AN ENTRY-LEVEL COBOL MAIN PROGRAM

Entry-Level COBOL programs can call FORTRAN subroutines and conversely. This enables a COBOL application to utilize the features of the FORTRAN language, such as the intrinsic routines, and FORTRAN run-time libraries.

The COBOL main program must be linked with all the called FORTRAN routines to form one bound unit. The FORTRAN routines and libraries must either be in the working directory or one of the libraries searched by the Linker, as specified by the Linker LIB and LIBn directives.

Figure 7-3 is a sample Entry-Level COBOL source program, COBFRT, whose function is to calculate and print the square roots of three integers. Since the COBOL library does not have a square root routine, a FORTAN subroutine, FRTRAN in Figure 7-4, is used to convert the passed COBOL integer argument values to read values and call the FORTRAN square root routine.

The commands entered from the operator terminal are listed in Figure 7-5. COBFRT.O and FRTRAN.O are both in the working directory FRTCOB, the COBOL run-time library, ZCRT, is in the directory specified by the Linker directive LIB, and the FORTRAN run-time library, ZFRT, is in the directory specified by LIB2. The system volume, ZSYS51, contains the FORTRAN and COBOL compilers, ZFRT, ZCRT and the operating system software. Volume FRTCOB contains the source modules (COBFRT.C and FRTRAN.F), the object modules (COBFRT.O and FRTRAN.O) and the linked bound unit COBFRT.

HONEYWELL INFORMATION SYSTEMS

Technical Publications Remarks Form

TITLE

CUT ALONG LINE

SERIES 60 (LEVEL 6) GCOS 6 MOD 400 PROGRAMMER'S GUIDE ADDENDUM A

ORDER NO.

CB22A, REV. 0

DATED JUNE 1978

ERRORS IN PUBLICATION

SUGGESTIONS FOR IMPROVEMENT TO PUBLICATION

••••••••••••••••••••••••••••••••••••••				
Your comments will be promptly invested as required. If you require a written ref	tigated by appropriate technic ply, check here and furnish co	cal personnel and action v omplete mailing address b	will be taken	
FROM: NAME			DATE	
TITLE			_	
COMPANY			_	
ADDRESS			_	

PLEASE FOLD AND TAPE – NOTE: U. S. Postal Service will not deliver stapled forms

PERMIT NO. 39531 WALTHAM, MA 02154 Postage Stamp Not Necessary if Mailed in the United States Postage Will Be Paid By: HONEYWELL INFORMATION SYSTEMS 200 SMITH STREET WALTHAM, MA 02154

ATTENTION: PUBLICATIONS, MS 486

Honeywell

CUT ALONG LIN

FOLD ALONG LINE

FIRST CLASS