SIMCON CDC VERSION 2

IMPLEMENTOR'S GUIDE AND

REFERENCE MANUAL

March, 1982
Updated February, 1983
Sea Grant Publication No. ORESU-H-83-001

E. L. Beals
Department of Fisheries and Wildlife
Oregon State University
Corvallis, Oregon 97331

TABLE OF CONTENTS

INTRODUCTION

SIMCON CDC Version 2 is written in FORTRAN IV and is a modified and extended version of the program developed originally for an IBM 370 computer at the Institute of Animal Resource Ecology, University of British Columbia, Vancouver, B.C., by Ray Hilborn, Bill Webb and Jeff Stander in 1976. This version can be implemented without modification on Control Data Corporation (CDC) 170 series, 70, and 6000 series computers under the Network Operating System (NOS).

This document serves as an aid to the implementation and maintenance of SIMCON on general computer systems and as a reference guide to the SIMCON algorithm and subprogram functions. The reader of this guide should have available a copy of the user's guide to the SIMCON language for SIMCON CDC Version 2 entitled SIMCON - A Simulation Control Language at Oregon State University, 1981, Sea Grant Publication No. ORESU-H1-83-001.

This work is a result of research sponsored by NOAA Office of Sea Grant, Department of Commerce, under Grant No. NA79AA-D-00106 (Project No. R/OPF-1). The U.S. Government is authorized to produce and distribute reprints for governmental purposes notwithstanding any copyright notation that may appear hereon.

METHOD AND CONDITIONS OF DISTRIBUTION

SIMCON CDC Version 2 is available without fee from:

Eric A. Rexstad
Department of Fisheries and Wildlife
Oregon State University
Corvallis, Oregon 97331

and is distributed with the condition that any published material reflect the original sources of SIMCON.

SIMCON versions and documents designed for IBM, DEC, and other machines are available as well from the University of British Columbia. For exact information regarding these other versions, write to:

Dr. Ray Hilborn
Institute of Animal Resource Ecology
University of British Columbia
Vancouver, B.C. VGT 1W5
Canada

Persons requesting SIMCON CDC Version 2 should provide a 600 ft. magnetic tape and specify the tape format as completely as possible. SIMCON CDC Version 2 will consist of 12 files with the following information. (The SIMCON source code files are described in more detail on page 13.)

File 1 : SIMCON source code: zero level overlay segment (0,0), 979 lines.
File 2 : SIMCON source code: overlay segment (1,0), 600 lines.
File 3 : SIMCON source code: overlay segment (1,1), 2261 lines.
File 4 : SIMCON source code: overlay segment (1,2), 513 lines.
File 5 : SIMCON source code: overlay segment (2,0), 122 lines.
File 6 : SIMCON source code: overlay segment (3,0), 127 lines.
File 7 : SIMCON symbol table data, 7 lines.
File 8 : SIMCON library source deck, 346 lines.
File 9 : Model test program, 76 lines.
File 10: Test program COMMON file, 2 lines.
File 11: Checkout procedure, 142 lines.
File 12: Checkout procedure verify output, ~400 lines.

In general, the information on tape can be sent in the following ways:

7 track	parity:	even
	density:	556 or 800 bpi
	character set:	CDC External BCD
	label options:	unlabelled
	blocking:	any size, 80 column card image format
	file separators:	imbedded tape marks
9 track	parity:	odd
	density:	800 or 1600 bpi
	character set:	EBCDIC or ASCII
	label options:	unlabelled or standard ANSI
	blocking:	any size, 80 column card image format
	file separators:	imbedded tape marks or ANSI end-of-file labels

For CDC users, information can be sent uncoded (binary with odd parity) in internal display format that can be directly transcribed to mass storage files.

FORTRAN IV GENERAL COMPILER DEPENDENCIES

SIMCON CDC Version 2 is written in ANSI 66 Standard FORTRAN except for routines performing specialized functions for which there are no ANSI 66 standards and for a few commonly found non-ANSI usages which are included in the list below. Specialized non-standard functions are organized into a few routines to permit easier substitutions by local routines. All non-standard functions are documented in this guide.

Certain features not always found on FORTRAN IV compilers are essential for the operation of SIMCON. Among the special properties are the following:

1. Labelled and blank common blocks must be available and it must be possible for some occurrences of a common block to be shorter than the first.

2. The number of words (or bytes) required to hold a real number must be at least as many as for an integer.

3. It must be possible to store an integer in a real variable and later extract it as an integer via equivalenced integer and real variables.

4. It must be possible to access variables in a common block by subscripting into an array whose first element is equivalenced to the first element of the common block.

5. It must be possible to pass an arbitrary array element to a subroutine to be treated as if it were the first element of the formal array argument. For example, the fourth element of the array BUFF is passed to a subroutine which acts as if that element represents the first element of the dummy array.

6. The compiler should accept the type identifier "INT EGER" (including the blank as it is shown) for the variable type "INTEGER". This is a convention used to identify varibles that were originally "LOGICAL*1" type on some IBM and PDP systems. Most often, (but not exclusively) these variables serve to manipulate character data.

7. If it is not otherwise possible to directly compare character data stored in A1 format, there must be some mechanism for doing this such as an escape to assembler language or other local routines. Direct character comparisons are made exclusively by one primitive level function subprogram.

8. There must be some means of performing indexed read and write operations on mass storage files, either indexed sequential or random access.

9. Hollerith constants must be allowed in data statements. All Hollerith constants in SIMCON are of the form "1Ha".

10. It must be possible to dimension a variable in a type or common statement. There are very few DIMENSION statements in SIMCON.

11. Logical type functions and variables must be supported.

12. G format conversions should be available. If not, a few output routines must be changed to use E or F formats.

13. Subscript expressions containing references to subscripted variables or 2 or more variable elements should be permitted.

14. Frequent use is made of DATA statements to define character arrays. It is expected that the order of a data item in the list will imply the array element to be assigned (the first character from the left is assigned to element 1, the second is assigned to element 2, etc.). Subscripts or implied DO loops within DATA statements are not used.

15. This version of SIMCON is very nearly the full operating version on the CDC CYBER at Oregon State University. It is not expected that the completely compiled, linked, and loaded program will fit all at once in core memory. Therefore, there must be a means of splitting the loaded program into sections or overlays. SIMCON is distributed as separate files with the anticipation that each file of subprograms may form an overlayed program segment.

COMMENTS ON UPGRADING TO FORTRAN V

In order to upgrade this version of SIMCON most easily, the FORTRAN V compiler should have the following extension beyond the ANSI 77 standard as well as support all the features listed in the preceeding section: The FORTRAN V compiler must accept the use of Hollerith constants and input alphanumeric data into integer variables with the A1 format conversion in the same manner as the FORTRAN IV standard compiler. It will not be possible to directly substitute the "CHARACTER*1" type declaration statement in the place of the "INT EGER" specification.

SYSTEM DEPENDENT FUNCTIONS

For the greatest ease of conversion to other systems, system dependent functions and parameters have been isolated in as few routines as convenient. The following discussion lists classes of system dependent routines and describes their functions. The detailed calling sequences of the subprograms listed are not given here, however, they may be found in the alphabetical listing of routines in Appendix A.

Basic Character Manipulation:

As a standard means of character handling, an operational concept of the "byte" was devised. A "byte" is defined as the smallest, directly addressable unit of memory capable of storing a single character. On various computers a "byte" can have a variety of manifestations. The simplest and most straight forward implementation of this concept is to set one integer element equivalent to one "byte", often one full word of memory. The "byte" defined in this manner should operate successfully for all machines. Many machines, however, are capable of directly addressing units of memory smaller than an integer element and for these machines, the "byte" can be defined to exploit this property thereby utilizing space more efficiently. Once the "byte" is defined, a "byte" may store a single character or the "byte" can be used as the basic unit of addressable storage to index blocks of mixed type variables.

In the SIMCON source code, the type identifier "INT EGER" is the convention employed to designate variables whose elements are implied to be

"byte" sized. If the "byte" is defined equivalent to one integer element, the identifier "INT EGER" need not be edited; the FORTRAN IV compiler will ignore the space and interpret it as "INTEGER". If the "byte" is defined to a size smaller than an integer element, all occurrances of type "INT EGER" must be changed to an appropriate type (such as "LOGICAL*1" for an IBM 370 computer). The definition of the "byte" and other machine parameters are made in the BLOCK DATA subprogram described on page 9.

It is advised, however, that for a byte addressable machine, SIMCON be made to operate first with the "byte" defined as an integer element since this is the form running on the CDC CYBER. SIMCON CDC Version 2 has undergone considerable revision on the CYBER since its arrival in 1976 and it may be that in some places, the "INT EGER" specification has not always been faithfully transcribed.

The following subprograms manipulate characters at the most primitive level. They are system dependent and will need to be re-written to fit each local implementation. Refer to the alphabetical listing of routines in Appendix A for the exact calling sequences and parameter definitions.

LOGICAL FUNCTION EQC

> compares a single character stored in one byte in A1 format with another character stored in one byte. EQC returns the value .TRUE. if they are equal and .FALSE. if they are not. EQC is considered system dependent since some machines cannot directly compare character data and must resort to intrinsic Boolean functions or assembly language.

SUBROUTINE PBYTE

moves one character stored in one "byte" in A1 format to a specified bit position within an integer element. It is expected that this function would be performed with local intrinsic Boolean functions such as MASK, AND, SHIFT, etc. This subroutine and subroutine GBYTE recognize that particularly for word addressable machines, characters stored in integer sized "bytes" are not usually space efficient and it would therefore be desirable to store large amounts of character data in a more compact form.

SUBROUTINE GBYTE

retrieves a "packed" character from an integer element and stores it in one "byte" in A1 format.

Although it was originally intended that active character manipulation be done with "unpacked" characters and that "packing" serve only to store large amounts of character data economically, the temptation to manipulate packed characters directly proved too much to resist on the CYBER (unfortunately, perhaps, for users of other machines). Thus, routines parallel to the basic character manipulation routines were developed to manipulate packed characters. Most of these routines are of course machine dependent, however, the calling sequences are identical to their unpacked counterparts. Thus, for a machine whose representation of packed and unpacked characters are functionally equivalent (such as the IBM 370), the two types of routines could be coded identically or duplicate entry points can be defined for the original routines.

LOGICAL FUNCTION EQC2

analogous to EQC, EQC2 compares two packed characters for equality.

LOGICAL FUNCTION EQCMP2

analogous to EQCMP (EQCMP is a system independent routine), EQCMP2 compares two packed character strings for equality. This routine is provided in a system independent form using EQC2 to perform a character by character comparison. However, it has since been found (for a word addressable machine only!) that a ten to twenty percent reduction in average execution time can be realized by streamlining the code of EQCMP2 as completely as possible (such as eliminating the call to EQC2 and incorporating its code).

SUBROUTINE MOVEST

moves a substring consisting of packed characters from one area of memory to another. Its calling sequence is identical to MAVEC (a system independent routine which moves characters stored in "bytes"). This routine is provided in a system independent form using a character by character move with PBYTE and GBYPTE but this method is not efficient.

Machine Configuration and Parameters:

Certain parameters and common block lengths are dependent on the particular machine and on the definition of the "byte" (see page 6). All machine dependent parameters and common blocks are localized and defined in 5 block data subprograms or subroutines. If the local FORTRAN IV compiler does not allow more than one block data subprogram, some can be converted to subroutines or combined into one block data subprogram. Except for subroutine CONFIG, these routines must be loaded into the zero level overlay program segment of SIMCON (See the SIMCON Main Program and Overlays, page 13). Refer to Appendix B for the exact definitions of the parameters and common blocks listed.

BLOCK DATA (unnamed)

defines block lengths and machine dependent parameters in the following blocks:

/NBCM/ /ZZZZ/ /MPST/ /MKRF/ /WKSP/
/MET/ /CSTACK/ /ATCMDS/

SUBROUTINE CCOM

is a user defined library loaded subprogram to define the size of the blank common block.

BLOCK DATA DEFCMD

defines and initializes the SIMCON command list in the following blocks:

/CMDLST/ /CMDNUM/

BLOCK DATA HCOM

is intended as a library loaded subprogram which defines and sets parameters in the common block /KCC/.

SUBROUTINE CONFIG

sets a parameter in the common block /KCC/ at execution time (by an executable assignment statement). This routine is contained in the initialization overlay segment (1,2); it must not reside in the zero level overlay as it may interfere with the library loaded block data subprogram HCOM.

File Input and Output:

These routines perform I/O functions for indexed direct access files and other file status processing for which there are no standard methods in FORTRAN IV.

SUBROUTINE REED

inputs a command image or character string data from a sequential file into a "byte" array. REED is included here since it performs a test for an end-of-file condition which is non-standard.

SUBROUTINE UCOMIO

processes indexed read and write operations to a scratch file used to store blank common block images. All records read or written are of the same length.

SUBROUTINE MACIO

processes indexed read and write operations to a file used to store SIMCON command macros. Record lengths vary but can be standardized. (No macro is longer than the buffer used--nominally 2000 "byte" elements.)

General Systems Functions:

(See also comments contained in the individual routines for more aid in the local implementation.)

SUBROUTINE SYINIT

performs system dependent functions during initialization, usually opening and initializing files.

SUBROUTINE SYSFN

performs system dependent functions required during appropriate phases of SIMCON execution such as closing, re-opening, rewinding files, error recovery, etc.

LOGICAL FUNCTION SYSTAT

tests for relevant operating system conditions during execution such as attention interrupts or job origin.

Optional Routines for System Extended Functions:

SUBROUTINE SYSCMD can be designed as a small auxiliary command processor to perform operating system commands or any other useful system function upon request. SYSCMD is called for any command which is preceeded by the dollar sign ($). SYSCMD is provided as a dummy routine.

SUBROUTINE CMDFIL is used to process a special system request resulting from the SIMCON "READ" command to attach a file to the job and open it as a SIMCON "batch" or data file. This is not a straight forward operation for the CDC CYBER. Other systems may find it more appropriate to perform this operation in other existing routines such as SYSCMD or SYSFN. CMDFIL is provided as a dummy routine.

THE SIMCON MAIN PROGRAM AND OVERLAYS

The main program for SIMCON CDC Version 2 opens files with the program statement and switches between the overlay main routines which perform the major functions. Systems other than CDC (which do not use the program statement to open files) will perform these functions in subroutine SYINIT. The calls which result in the loading of the appropriate overlay program segment are accomplished by the CALL OVERLAY statement on the CDC CYBER. The comments in the main program describe the function of the overlay program segment being loaded at the time and the name of the particular routine where program control is actually being transferred in a manner similar to a subroutine call. The controlling algorithm has been carefully constructed such that two consecutive calls to the same overlay never occur; a return from a secondary overlay to the SIMCON main routine automatically implies that a new overlay is to be loaded.

Figure 1 illustrates the overlay structure of SIMCON and their relative positions in memory. The overlays are organized such that any routine may reference any other routine in the same overlay and any routine in an overlay loaded at a lower address in memory at the time. SIMCON does not require local common blocks or variables to be re-initialized to original values or set to their most recent values; all such variables are explicitly redefined each time a routine is called in a newly loaded overlay. All overlay communication is effected through global common blocks and variables contained in the zero level overlay which must never be unloaded from memory.

Switching control is accomplished through the flag variables in common block /OVTALK/. These switches are set only by main overlay routines. Refer to Appendix B for the definition of common block /OVTALK/ and these switches.

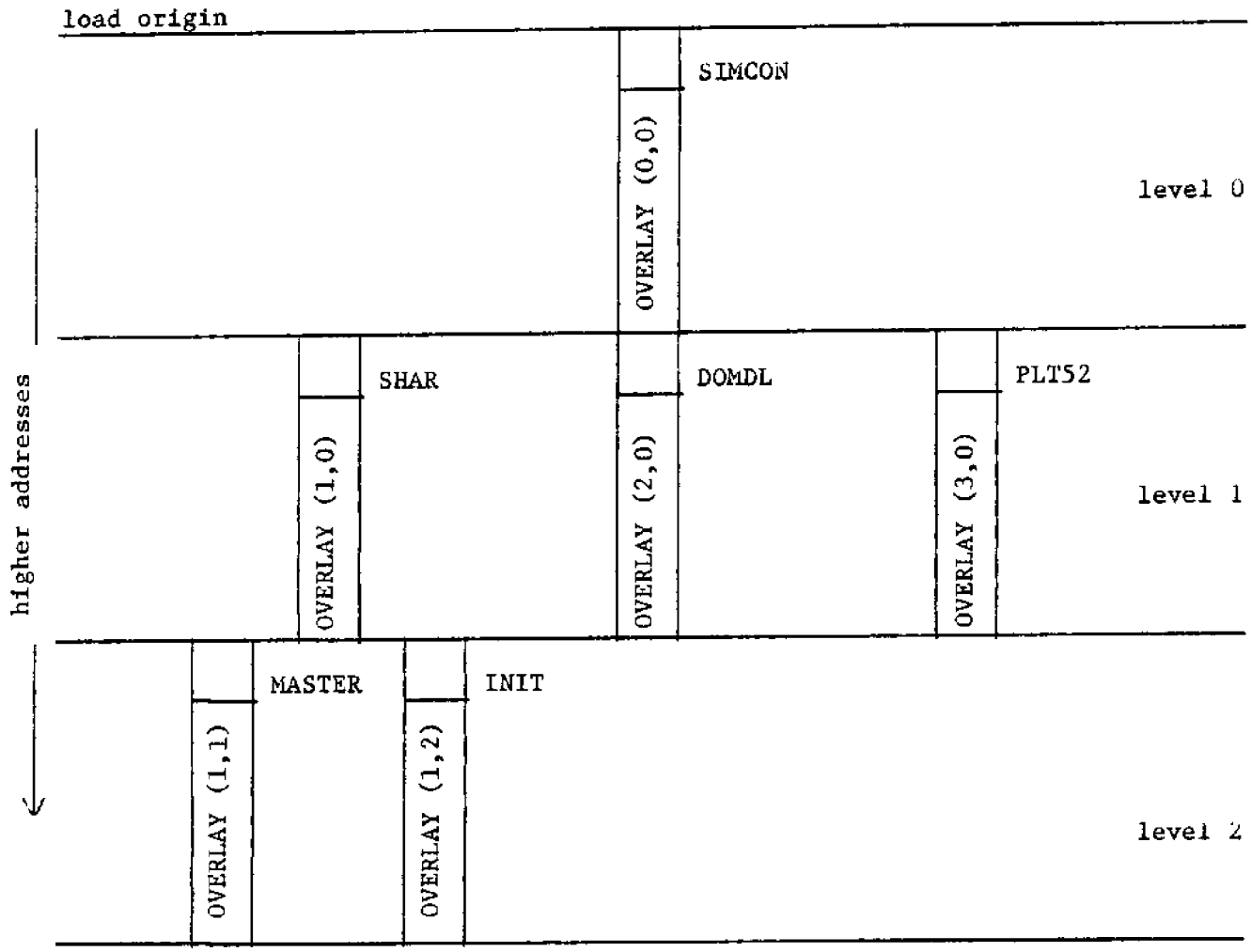

Figure 1. The overlay structure of SIMCON in memory. The main program (SIMCON) and OVERLAY (0,0) is loaded first and remains in memory throughout execution. All global common blocks reside here. At any given level, only one overlay can be loaded at one time and all program references are within the same overlay or to overlays loaded lower in memory concurrently. Routine SHAR in OVERLAY (1,0) explicitly causes the loading of either MASTER or INIT. During command input from a terminal, for example, the overlays containing SIMCON, SHAR and MASTER would be currently loaded in memory.

The SIMCON CDC Version 2 source code is organized into six overlay program segments in separate files. The first program unit in each file is the main overlay routine to which program control is transferred when the overlay is loaded. By CDC CYBER convention, each of these main overlay routines begin with a PROGRAM statement instead of SUBROUTINE. The first file contains the root or zero level overlay [OVERLAY (0,0)] which is loaded first and remains loaded during all phases of SIMCON execution. The zero level overlay contains utility routines and common blocks that are used by all other secondary overlays.

File 2 contains more utility routines used by other overlays. Routine SHAR is the main program unit for this overlay [OVERLAY (1,0)].

File 3 contains the routines responsible for the individual command functions and the master control and cycling program [OVERLAY (1,1)]. The master program (routine MASTER) requests user command input and invokes the appropriate command function. Routine MASTER makes the decisions of the overlays to be loaded when necessary and sets the proper switches for the SIMCON main program.

File 4 contains initialization routines executed only once. Routine INIT is the main program unit for this overlay [OVERLAY (1,2)].

File 5 contains the user model monitoring routines. The loaded overlay [OVERLAY (2,0)] also will contain the user model routines. Routine DOMDL is the main program unit.

File 6 separates the graphics functions from the main body of commands. Graphics functions will almost always have to be written locally for each computer system. The implementation of the graphics functions can be delayed until the major part of SIMCON is operating satisfactorily by providing a dummy subroutine in place of routine PLT52, the main program unit for this overlay [OVERLAY (3,0)].

Loading Considerations

Library searching -- the order of search should be:

1. the model and associated optional routines,
2. the SIMCON library, SIMLIB,
3. the FORTRAN and system libraries.

It is convenient from the user's point of view to library load certain subprograms that are not necessarily referenced (by CALL statements) in order to fix important COMMON blocks at user determined lengths at load time. If it is possible to instruct the linker-loader to load these routines into the zero-level overlay, do so. If not, fix these routines into the zero level overlay and define the block lengths and parameters to convenient defaults.

SUBROUTINE CCOM

> Library loaded from the user's model or the SIMCON library, CCOM defines the size of the blank COMMON block at load time. The blank COMMON block is reserved solely for the user's use. (See Part II of the SIMCON user's manual, "The FORTRAN Programmer's Guide", page 30.)

SUBROUTINE DRAND

> Library loaded from the SIMCON library. DRAND is a random number utility routine.

SUBROUTINE DRSET

> Library loaded from the SIMCON library, DRSET is a random number utility routine.

BLOCK DATA HCOM

> Library loaded from the SIMCON library, HCOM defines a COMMON block at load time. See pages 10, 19.

Figure 2.

The SIMCON loading procedure on the CDC CYBER 170 Model 720, Network Operating System (NOS). The basic steps are: making the files SUPCOM (SIMCON internal data), SIMLIB (SIMCON library), and SIMBIN (SIMCON object deck) local to the job; building a library file from the user model object deck (BMODL); instructing the loader to force the loading of CCOM, DRAND, RANSET (a random number generator from the FORTRAN library), and HCOM even though they do not necessarily satisfy external references; then defining the library search order. The absolute load module is placed on the file SIMCON (named on an OVERLAY directive in the SIMCON object deck). The field length required is approximately 27000, 60 bit words on the CYBER.

```
.PROC,SIM.
OFFSW(1)
IFE,DT=TXD,TIMESHARE,ORIGIN JOB
   ONSW(1)
ENDIF,TIMESHARE.
SETTL,30.
RETURN,SIMCON,QL.
GET,SUPCOM,SIMLIB/UN=AAVI7M.
ATTACH,SIMBIN/UN=AAVI7M.
LIBGEN,F=BMODL,P=QL.
LDSET(USE=CCOM/DRAND/RANSET,USEP=HCOM)
LDSET(LIB=QL/SIMLIB/OSULIB/COMPLOT)
LOAD(SIMBIN)
NOGO.
RETURN,QL,SIMLIB,SIMBIN.
REVERT. SIMCON GENERATION COMPLETE
$REVERT.CCL
/
```

AN IMPLEMENTATION CHECKLIST

1. Separate the files from the tape into separate files on mass storage, page 2. In particular, File 7, SIMCON symbol table data, call it "SUPCOM"; File 10, test program COMMON block, call it "COMMON"; and File 11, checkout procedure, call it "BATCH".

2. Obtain a complete listing of all files.

3. Edit the SIMCON source code to set parameters and COMMON block lengths appropriate for this installation:

   a. BLOCK DATA (unnamed), in overlay segment (0,0)

      i) Define the "byte", page 6: For the first attempt at running SIMCON set the "byte" equivalent to a full integer element (one word) by setting parameters in block /NBCM/. See the block description in Appendix B.

      ii) For all COMMON blocks in BLOCK DATA (unnamed), determine their lengths and parameter settings. Refer to the individual block descriptions in Appendix B.

   b. BLOCK DATA DEFCMD, in overlay segment (0,0)

      i) Set block lengths, parameters, and data for each block. Note that the command names are packed in sequential elements, each element comprising of the same number of full integers (in this case, 2 words holding 10 characters apiece, left justified, blank filled). See individual descriptions in Appendix B.

      ii) If named BLOCK DATA subprograms are not permitted by the local compiler, incorporate BLOCK DATA DEFCMD within BLOCK DATA (unnamed).

   c. BLOCK DATA HCOM, in the SIMCON library source deck

      i) Set block lengths and parameters. See Appendix B.

      ii) If named BLOCK DATA subprograms are not permitted, change BLOCK DATA HCOM to SUBROUTINE HCOM.

   d. SUBROUTINE CONFIG, in overlay segment (1,2) sets a parameter at execution time. See Appendix B.

4. Modify character manipulation routines. Modifications necessary will depend on the particular installation, see page 7. Refer to Appendix A for the function descriptions and calling parameters.

   a. LOGICAL FUNCTION EQC in overlay segment (0,0)

   b. SUBROUTINE PBYTE in overlay segment (0,0)

   c. SUBROUTINE GBYTE in overlay segment (0,0)

   d. LOGICAL FUNCTION EQC2 in overlay segment (1,0)

   e. LOGICAL FUNCTION EQCMP2 in overlay segment (1,0)

   f. SUBROUTINE MOVEST in overlay segment (1,1)

5. Modify system dependent I/O functions, page 11. Refer to Appendix A for the function descriptions and calling parameters.

   a. SUBROUTINE REED in overlay (1,0)

   b. SUBROUTINE UCOMIO in overlay (0,0)

   c. SUBROUTINE MACIO in overlay (0,0)

6. Modify general systems function routines, page 11. See especially the comments within the individual routines for aids to the necessary changes. Refer also to Appendix A.

   a. SUBROUTINE SYINIT in overlay segment (1,2)

   b. SUBROUTINE SYSFN in overlay segment (0,0)

   c. SUBROUTINE SYSTAT in overlay segment (0,0)

7. Overlay main routines, page 13. Modifications depend on the local installation, particularly the OVERLAY, CALL OVERLAY, and PROGRAM statements.

   a. PROGRAM SIMCON in overlay segment (0,0) (This is the main program.)

   b. PROGRAM SHAR in overlay segment (1,0)

c. PROGRAM MASTER in overlay segment (1,1)

d. PROGRAM INIT in overlay segment (1,2): Note that some system dependent parameters not mentioned previously are defined here and a non-standard end-of-file test is performed on unit 12 twice. See the comments in the source code for a guide. See also Appendix B, COMMON block /SIZCOM/.

e. PROGRAM DOMDL in overlay segment (2,0)

f. PROGRAM PLT52 in overlay segment (3,0): For the first attempt, temporarily short-circuit this overlay with a RETURN statement. This overlay exclusively produces output for graphic terminals or other devices and its modification can be deferred to a later time.

8. Instructions to the linker-loader, page 17.

   a. Establish the library search order.

   b. Determine the method to force-load the routines listed on page 17 from the libraries into the zero-level overlay if it is convenient.

9. Compile SIMCON, the SIMCON library, and the test model.

10. Link and load SIMCON

11. Run SIMCON: SIMCON will attempt to read the COMMON file and the checkout procedure BATCH.

12. Compare the output generated by the checkout procedure with the verify output provided.

THE SIMCON OPERATING ALGORITHM

1. Initialization

   a. Load zero level overlay to start.

   b. Load OVERLAY (1,0) and transfer control to routine SHAR.

   c. SHAR loads OVERLAY (1,2) (containing initialization routines).

   d. System and file initializations are performed and the SIMCON symbol table is created.

   e. Return control to routine SHAR which loads OVERLAY (1,1) and transfers control to MASTER.

2. Determine command entry mode. MASTER has two entry points controlled by switch ISWA, the normal mode and the intervention mode. The normal mode is selected at initialization and begins at step 3. The intervention mode begins at step 4.

3. Enter normal command mode.

   a. Request from the command stack the next command pending. (At initialization, the stack is empty.) If the stack is not empty, pass command fetched to the command pre-processor at step 5. If the stack is empty, continue with the next step.

   b. Request a command from the current input unit. At initialization, the current unit is the "batch" input file rather than the unit assigned to the terminal.

   c. Pass command fetched to the command pre-processor.

   d. If an end-of-file is encountered on the "batch" file, set current input to the unit assigned to the terminal and repeat at 3b. If an end-of-file is encountered on the unit normally assigned to a terminal, process according to job origin (stop if batch origin, repeat at 3b if time share origin). If the local system does not generate E-O-F's from an interactive terminal, no change of the source code should be necessary.

4. Intervention Command Processing. This entry mode processes commands which have been designated by the user to intervene at some point of model iteration ("AT" commands). Immediately before this mode is entered, the model overlay was loaded and the model was in the process of iteration. At some point, the model monitor detects a command or commands designated by the user to intervene. The monitor places these commands into the stack and signals the overlay main routine (DOMDL) to return control to the master overlay and enter via the intervention point.

   a. Request from the command stack the next intervention command pending and pass it to the command pre-processor at step 5.

   b. If the stack stop-point has been encountered (a device used to separate intervention from normal commands and set by the model monitor), clear stop point and return control to the model.

   c. Processing finished. Return control at step 2.

5. Command Pre-processor

   a. Scan for a comment line (one of two types) and if the command is a comment, pass control to the command executor (which echos the line or does nothing as the case may be).

   b. Scan for command continuation on the next line (indicated by a terminating comma). If command is continued, request next line from current input unit and repeat at 5b.

   c. Scan for multiple commands on one line. If more than one command is contained on the line, retain the first and push the rest onto the command stack in reverse order of their execution sequence (from right to left). Now pass command retained to the command executor at step 6.

6. Command Executor

   a. Identify the command

   b. Execute command. If the command requires that the model, graphics, or initialization overlays be loaded, set the appropriate switch (ISWB or ISWC) and return control through routine SHAR to the SIMCON main program which will load the overlay. When these overlays have completed their functions, they return control to the SIMCON main program which will reload the MASTER overlay.

   c. Processing finished. Return control at step 2.

APPENDICES

Appendix A: Alphabetical list of routines and calling parameters.

Disclaimer. Although considerable effort has gone towards maintaining the accuracy of the following lists, some errors undoubtedly have crept in. Of course, if discrepancies do occur, defer on the side of the actual source code.

ATMGR (IBUFF,LEN,IT,IFN,IRET)

Subroutine.
Handles the detail of the AT command table mechanics.

IBUFF	A string array containing a command image in A1 format.
LEN	The number of characters in IBUFF.
IT	An iteration reference value.
IFN	A function code: 1 Store the command into the AT list. 2 Fetch all AT commands whose effective iteration matches the reference value IT and load them into the command stack for execution. 3 Clear the AT list.
IRET	Return condition code: 0 Operation complete. 1 No commands match IT or list empty. 2 List is full. 3 Stack manager error.

Referenced by	from Overlay
ATS	(0,0)
INIT	(1,2)
RTSTUF	(2,0)

External References to	in Overlay
MOVEW	(0,0)
STKMGR	(0,0)
PACKC	(0,0)

Common Blocks Used	Defined by	in Overlay
/ATCMDS/	BLKDAT	(0,0)
/SUPCOM/	SIMCON	(0,0)
/WKSP/	BLKDAT	(0,0)

ATS (IBUFF,LEN)

Subroutine.
Processes an AT command.

IBUFF	Contains a command image in A1 format.
LEN	The length of Ibuff.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
ATMGR	(0,0)
EQCMP	(0,0)
FGET	(0,0)
SCAN	(1,0)
SCSET	(1,0)
SYSTAT	(0,0)
UNPACK	(0,0)

Common Blocks Used	Defined by	in Overlay
/ATCMDS/	BLKDAT	(0,0)
/SUPCOM/	SIMCON	(0,0)
/WKSP/	BLKDAT	(0,0)

ATTN

Subroutine.
A user callable routine designed to interrupt model iteration conditionally. ATTN provides a means for a model programmer to trap errors within the model. This routine is not called by any SIMCON subprogram.

Common Blocks Used	Defined by	in Overlay
/SYSCOM/	SYSFN	(0,0)

BLOCK DATA (unnamed)

Block Data Subprogram.
Most common blocks and parameters that may need to vary from machine to machine are defined here. Refer to the idividual common block descriptions for more details.

Common Blocks Used	Defined by	in Overlay
/ATCMDS/	BLKDAT	(0,0)
/CSTACK/	BLKDAT	(0,0)
/MET/	BLKDAT	(0,0)
/MKRF/	BLKDAT	(0,0)
/MPST/	BLKDAT	(0,0)
/NBCM/	BLKDAT	(0,0)
/WKSP/	BLKDAT	(0,0)
/ZZZZ/	BLKDAT	(0,0)

CCOM

Subroutine.
Optional user supplied subroutine to fix the size of the blank common block. A default CCOM fixing blank common at 1000 words is otherwise loaded from the SIMCON library. This subroutine should have no executable statements other than a RETURN statement as CCOM is never actually called.

CHANGE (OLD,OL,NEW,NL,OLINE,OLEN,NLINE,NLEN,IRT)

Subroutine.
A text editing routine used for parameter substitutions during macro processing. This routine operates on packed character strings but is system independent.

OLD	An integer array containing a symbolic parameter in packed form.
OL	The number of characters contained in OLD.
NEW	An integer array containing a symbolic parameter to replace OLD (in packed form).
NL	The number of characters contained in NEW.
OLINE	An integer array containing the string to edit in packed form.
OLEN	The number of characters contained in OLINE.
NLINE	An integer array returned containing the edited string in packed form.
NLEN	The number of characters returned in NLINE.

IRT — A return condition code:
- 0 Editing complete.
- 1 No occurance of parameter OLD found.

Referenced by	from Overlay
MACROS	(1,1)

External References to	in Overlay
EQCMP2	(0,0)
MOVEST	(1,1)
SCAN2	(1,1)
SCSET	(1,0)

CMDFIL (IBUFF,LEN,IERR,IRET)

Subroutine.
A machine dependent routine designed to process the SIMCON READ command. It should function by opening a new BATCH file specified by the command. Provided here only as a dummy routine.

Referenced by	from Overlay
MASTER	(1,1)

CMREAD (IN,ICOM,LENGTH)

Subroutine.
Constructs the symbol table from the source definitions of user and SIMCON's utility common blocks.

IN — The logical input unit number of the common block definition file.

ICOM — A code for the common block to process:
- 1 The SIMCON utility block, SUPCOM.
- 2 The blank common block.

LENGTH — The total length of the common block in "bytes" (see the definition of "byte" in the description of common block /NBCM/.

Referenced by	from Overlay
INIT	(1,2)

External References to	in Overlay
CONROT	(1,2)
EQCMP	(0,0)
FANDC	(1,0)
RDSTMT	(1,2)
SCAN	(1,0)
SCSET	(1,0)

Common Blocks Used	Defined by	in Overlay
/COMCOM/	CMREAD	(1,2)
/NBCM/	BLKDAT	(0,0)
/SIZCOM/	VARMV	(0,0)

CONFIG

Subroutine.
Sets machine parameters at execution time.

Referenced by	from Overlay
INIT	(1,2)

Common Blocks Used	Defined by	in Overlay
/KCC/	BLKDAT	(0,0)

CONROT (ISTR,IL,ITYPE)

Subroutine.
A symbol table initialization utility routine. Resposible mainly for interpreting subscripts and computing common block displacements.

ISTR	Contains a variable name in A1 format.
IL	The number of characters in ISTR.
ITYPP	A code for the specification statement currently being processed.

1 INTEGER
2 INTEGER*2
3 REAL
4 LOGICAL
5 LOGICAL*1
6 COMMON
7 DIMENSION

Referenced by	from Overlay
CMREAD	(1,2)

External References to	in Overlay
EQC	(0,0)
FANDC	(1,0)
SUBDET	(1,0)
TASET	(1,2)

Common Blocks Used	Defined by	in Overlay
/COMCOM/	VARMV	(0,0)
/SIZCOM/	VARMV	(0,0)

DEFCMD

Block Data Subprogram. Sets parameters and initializes data for the SIMCON command repertoire.

Common Blocks Used	Defined by	in Overlay
/CMDLST/	BLKDAT	(0,0)
/CMDNUM/	BLKDAT	(0,0)

DELT (IKEY,IDATA,IRET)

Subroutine.
Deletes an entry from the symbol table.

IKEY	A variable name.
IDATA	Variable attributes returned.
IRET	Return condition code: 1 Entry deleted. 2 Entry not found.

Referenced by	from Overlay
IHSH	(1,0)

External References to	in Overlay
FINDH	(1,0)

Common Blocks Used	Defined by	in Overlay
/KCC/	HCOM	(0,0)

DOMDL

Overlay (2,0) Main Program.
Cycles between the model monitor routine RTSTUF and the user's model.

Referenced by	from Overlay
SIMCON	(0,0) via CALL OVERLAY

External References to	in Overlay
RTSTUF	(2,0)
STKMGR	(0,0)
STPRT	(0,0)
STZER	(0,0)
SYSTAT	(0,0)
UMODEL	(2,0)

Common Blocks Used	Defined by	in Overlay
/OVTALK/	SIMCON	(0,0)
/SUPCOM/	SIMCON	(0,0)
/XPLT52/	LPLT5	(0,0)

DZB (IFROM,INT,NC,NSD,IFIL,IRET)

Subroutine.
Translates variable subscripts from characters to integer values.

IFROM	A subscript character string.
INT	An integer subscript value returned.
NC	The length of IFROM.
NSD	Unused.
IFIL	A delimiting character (usually a comma).
IRET	Return condition code: 1 Subscript value returned.

2 Syntax error.

Referenced by	from Overlay
SUBDET	(1,0)

External References to	in Overlay
FGET	(0,0)
MAVEC	(0,0)
SCAN	(1,0)
SCSET	(1,0)

ENTR (IKEY,IDATA,IRET)

Subroutine.
Enters a variable name and its attributes into the symbol table.

IKEY Variable name.
IDATA The variable attributes (see common block /COMCOM/ for a description of variable attributes).
IRET Return condition code:
1 Variable entered.
2 Table overflow.

Referenced by	from Overlay
IHSH	(1,0)

Referenced by	from Overlay
PACKC	(0,0)

Common Blocks Used	Defined by	in Overlay
/KCC/	HCOM	(0,0)

EQC (CH1,P1,CH2,P2)

Logical Function.
Tests for equality two characters in A1 format.

CH1	Starting position of a string.
P1	Character position within string CH1.
CH2	Starting position of the second string.
P2	Character position within string CH2.

Referenced by	from Overlay
CONROT	(1,1)
EQCMP	(0,0)
FANDC	(1,0)
FGET	(0,0)
IGNORE	(1,0)
LPLT51	(0,0)
LSIZEB	(0,0)
MASTER	(1,1)
RDSTMT	(1,2)
WHICHC	(1,1)

EQC2 (ISTR1,IPOS1,ISTR2,IPOS2)

Logical Function.
A character comparison routine for characters in packed strings analogous to subroutine EQC.

ISTR1	First word address of a packed character string.
IPOS1	Position of a character counted from the left most position of the first word as position number 1. IPOS1 should not be constrained by the number of characters that may fit into an integer for it should be able to take on any positive number.
ISTR2	First word address of a packed character string.
IPOS2	Position of a character in ISTR2.

Referenced by	from Overlay
EQCMP2	(1,0)
SCAN2	(1,0)

EQCMP (S1,P1,S2,P2,LEN)

Logical Function.
Tests two strings in A1 format for equality.

S1	Starting address of a packed string.
P1	Character position within the string S1.
S2	Starting postion of a second string.
P2	Character postion within string S2.
LEN	The number of characters to compare.

Referenced by	from Overlay
ATS	(1,1)
CMREAD	(1,2)
FANDST	(1,0)
MASTER	(1,1)

External References to	in Overlay
EQC	(0,0)

EQCMP2 (ISTR1,IPOS1,ISTR2,IPOS2,LEN)

Logical Function.
Tests for equality two packed character strings.

ISTR1	First word address of a packed character string.
IPOS1	Position of the first character of the substring to test counted from the left most position of the first word as position number 1. IPOS should not be constrained by the number of characters that may fit into an integer for it should be able to take on any positive number.
ISTR2	First word address of a packed character string.
IPOS2	Position of the first character of the substring to test.
LEN	The number of characters to compare.

Referenced by	from Overlay
CHANGE	(1,1)
FINDH	(1,0)
GETMAC	(1,1)
MACROS	(1,1)
NAMCMD	(1,1)
PUTMAC	(1,1)
WHICHC	(1,1)

External References to	in Overlay
EQC2	(0,0)

FANDC (IARRAY,LEN,ICHAR,NUM,IST,IFIN,ICF,IRET)

Subroutine.
Finds in a string IARRAY an occurance of any character in the string ICHAR.

IARRAY	A character string in A1 format.
LEN	The number of characters in IARRAY.
ICHAR	A character string in A1 format.
NUM	The number of characters in ICHAR.
IST	The character position in IARRAY to begin the search.
IFIN	Position of the first matching character in IARRAY returned.
ICF	The position of the matching character in ICHAR returned.
IRET	Return condition code: 1 A matching character was found. 2 No match found. 3 A parameter error, IST,NUM <= 0 or IST > len.

Referenced by	from Overlay
CMREAD	(1,2)
CONROT	(1,2)
MACROS	(1,1)
NAMVAR	(1,1)
SCAN	(1,0)
STGT	(1,1)
STRFND	(1,1)
VGTS	(1,1)
WHICHC	(1,1)

External References to	in Overlay
EQC	(0,0)

FANDST (IARRAY,LEN,ISTR,NUMB,IS,IF,IRET)

Subroutine.
Finds the position of a substring in a string.

IARRAY	A character string in A1 format.
LEN	The number of characters in IARRAY.
ISTR	The character substring in A1 format.
NUMB	The number of characters in ISTR.
IS	The position of the character in IARRAY to begin the search.
IF	The position of the first character where ISTR is found in IARRAY.
IRET	Return condition code: 1 ISTR found at position IS. 2 ISTR not found. 3 Parameter error, IS,NUMB= 0 or IS<=LEN.

Referenced by	from Overlay
LOOPER	(1,1)
NAMVAR	(1,1)
SIMLT	(1,1)
STATS	(1,1)
STRFND	(1,1)
SUBDET	(1,0)
VALFND	(1,0)
VGTS	(1,1)
VIEWX	(1,1)

External References to	in Overlay
EQCMP	(0,0)

FGET (ICARD,I1,I2,X,IERR)

Subroutine.
Converts a character string representing a numeric value into the internal floating point representation. It is also called from routines requesting an integer value where the floating point number is converted to integer. General floating point syntax is accepted including E specifications.

ICARD	A character string in A1 format.
I1	First character position of the substring to convert.
I2	Last character position of the substring to convert.
X	The value converted returned.

IERR Return condition code:
0 value successfully converted.
1 syntax error or characters are non-numeric.

Referenced by	from Overlay
ATS	(1,1)
DZB	(1,0)
SIMLT	(1,1)
TIMEF	(1,1)
VALFND	(1,0)
VGTS	(1,1)

External References to	in Overlay
EQC	(0,0)

FILE (IBUFF,LEN,COMON1)

Subroutine.
Proccesses the FILE command.

IBUFF A command image in A1 format.
LEN The length of IBUFF.
COMON1 Unused.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
VIEWX	(1,1)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

FINDH (IKEY,IDATA,IENT,IRET)

Subroutine.
Looks up a variable name in the symbol table and returns the entry number and variable attributes.

IKEY	An array containing the variable name in A1 format.
IDATA	Variable attributes returned (see common block /COMCOM/ for details).
IENT	The entry number returned.
IRET	Return condition code: 1 Entry found. 2 Entry not found.

Referenced by	from Overlay
DELT	(1,0)
IHSH	(1,0)

External References to	in Overlay
EQCMP2	(0,0)
PACKC	(0,0)

Common Blocks Used	Defined by	in Overlay
/KCC/	BLKDAT	(0,0)
/WKSP/	BLKDAT	(0,0)

GBYTE (IBUFF,POS,CHAR)

Subroutine.
Gets a character from a packed string.

IBUFF	An integer array containing a character string in packed form.
POS	The character position counting from the left most position of the first word of IBUFF as position number 1.
CHAR	One character in A1 format returned.

Referenced by	from Overlay
MOVEST	(1,1)
UNPACK	(0,0)

GETMAC (NAME,LN,IBUFF,LEN,IRET)

Subroutine.
Retrieves a macro from the macro library.

NAME	The name of a macro in A1 format.
LN	The length of NAME.
IBUFF	An integer array returned containing the macro text.
LEN	The length of the integer array IBUFF.
IRET	Return condition code: 0 macro found and returned in IBUFF. 1 macro not found or library empty.

Referenced by	from Overlay
MACROS	(1,1)

External References to	in Overlay
EQCMP2	(1,0)
MACIO	(0,0)
MAVEC	(0,0)
PACKC	(0,0)
SETC	(0,0)

Common Blocks Used	Defined by	in Overlay
/MET/	BLKDAT	(0,0)
/WKSP/	BLKDAT	(0,0)

GETVAR (IBUFF,LEN)

Subroutine.
Proccesses the GET command.

IBUFF	A command image in A1 format.
LEN	The length of IBUFF.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
VIEWX	(1,1)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

GRAPH (IBUFF,LEN,COMON1)

Subroutine.
Processes the GRAPH command.

IBUFF	A command image in A1 format.
LEN	The length of IBUFF.
COMON1	Unused.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
VIEWX	(1,1)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

HCOM

Block Data Subprogram.
An optional user supplied subprogram to fix the sixe of the symbol table. By default, the symbol table is fixed to 2200 words.

Common Blocks Used	Defined by	in Overlay
/KCC/	HCOM	(0,0)

HDUMP (IPAR)

Subroutine.
Outputs an image of the current symbol table listing variable names, their attributes, common block residence, and displacements.

IPAR — Output type code:
- 0 Output table for common block /SUPCOM/ and blank common.
- 1 Blank common only.
- 2 /SUPCOM/ only.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
SYSTAT	(0,0)
UNPACK	(0,0)

Common Blocks Used	Defined by	in Overlay
/KCC/	BLKDAT	(0,0)

IGNORE (BUFF,LEN,CHARS,LCHAR,INPTR,LAST,IRET)

Subroutine.
The compliment to FANDC, IGNORE finds the first occurrence of a character in string BUFF not in the list CHARS.

BUFF — A character string in A1 format.
LEN — The number of characters in BUFF.
CHARS — The list of characters to ignore in A1 format.
LCHAR — The number of characters in CHARS.
INPTR — The character position within BUFF to begin the search.
LAST — The position within BUFF of the first character not in the CHARS list.
IRET — Return condition code:
- 1 Non-matching character was found.
- 2 All characters are in the CHARS list.
- 3 Parameter error, INPTR > LEN.

Referenced by	from Overlay
VALFND	(1,0)

External References to	in Overlay
EQC	(0,0)

IHSH (IF,IKEY,IDATA,IRET)

Subroutine.
Translates a symbol table manipulation into calls to one or more basic symbol table routines.

IF	Function code: 1 Enter new entry. 2 Lookup existing entry. 3 Lookup entry and if not found, enter given entry. 4 Delete old entry. 5 Replace existing entry with given entry.
IKEY	Entry name in A1 format.
IDATA	Entry attributes (see common block /COMCOM/ for description).
IRET	Return condition code: 1 Operation complete. 2 entry not found or table overflow.

Referenced by	from Overlay
NAMVAR	(1,1)
STGT	(1,1)
TASET	(1,2)

External References to	in Overlay
DELT	(1,0)
ENTR	(1,0)
FINDH	(1,0)

INIT

Overlay (1,2) Main Program.
Main program for the initialization routines which construct the symbol table, initialize common blocks, and perform user defined initializations.

Referenced by	from Overlay
SHAR	(1,0) via CALL OVERLAY

External References to	in Overlay
ATMGR	(0,0)
CMREAD	(1,2)
CONFIG	(1,2)
INITCM	(1,2)
INITER	(1,2)
STKMGR	(0,0)
SYINIT	(0,0)
SYSFN	(0,0)
UINIT	(1,2)

Common Blocks Used	Defined by	in Overlay
/KCC/	BLKDAT	(0,0)
/NBCM/	BLKDAT	(0,0)
/OVTALK/	SIMCON	(0,0)
/SIZCOM/	VARMV	(0,0)
/SUPCOM/	SIMCON	(0,0)

INITCM

Subroutine.
Zeros the blank common block.

Referenced by	from Overlay
INIT	(1,2)
MASTER	(1,1)

Common Blocks Used	Defined by	in Overlay
//	CCOM	(0,0)
/KCC/	BLKDAT	(0,0)

INITER

Subroutine.
Initializes the SIMCON utility block.

Referenced by	from Overlay
INIT	(1,2)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

LOOPER (INBUFF,LEN,IC)

Subroutine.
Processes the SET, DISPLAY, PLOT, and UNPLOT commands.

INBUFF	A command image in A1 format.
LEN	The length of INBUFF
IC	The command number of the function to perform. (See the comments in routine MASTER for a table of command numbers.)

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
FANDST	(1,0)
MAVEC	(0,0)
RPUTS2	(1,1)
RTAKE2	(1,1)
STGT	(1,1)
STRFND	(1,1)
SYSTAT	(0,0)
VALFND	(1,0)
VARMV	(0,0)
VGTS	(1,1)
VPRNT	(1,1)

Common Blocks Used	Defined by	in Overlay
/HELP/	STATS	(0,0)
/SUPCOM/	SIMCON	(0,0)
/SIZCOM/	VARMV	(0,0)

LPLT5 (Y,YMAX,ITIME,IYB,NPAR,NLLIN,LOGU,IWGRA,IFN)

Subroutine.
Switches between graphic and printed output routines for simulation variables.

IFN	A function code: 1,2,3 Save values then pass all parameters directly to subroutine LPLT51 for printed output. (See LPLT51 for descriptions of the parameters.) 3,4 Save values then set flags for subsequent calling of the graphics routines. 6 Save values only (the GET function).

Referenced by	from Overlay
RTSTUF	(2,0)
VIEWX	(1,1)

External References to	in Overlay
LPLT51	(0,0)
SYSTAT	(0,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)
/XPLT52/	SIMCON	(0,0)

LPLT51 (Y,YMAX,ITIME,IYB,NPAR,NLLIN,LOGU,IWGRA,IFN)

Subroutine.
The "dispose" routine for formatted output and printer plots.

Y	An array containing a line of Y values to be printed or plotted.
YMAX	An array containing scaling maximums for the values in Y.
ITIME	The current iteration in model "years".
IYB	A reference value: if ITIME = IYB, produce the coordinate axes or table headers as appropriate.
NPAR	The number of values in array Y.
NLLIN	The number of printer columns to use on the page.
LOGU	The logical unit number to receive output.
IWGRA	A function code: 1 The output values represent variables currently in the plotting queue.

2 The output values represent variables named on the command.

IFN A function code:
1 VIEW command output (printer plot)
2 PRINT command output (printed table)
3 Reserved for graphics.
4 Reserved for graphics.
5 FILE command output (printed table written to a file without the header).

Referenced by	from Overlay
LPLT5	(0,0)

External References to	in Overlay
EQC	(0,0)
LSIZEB	(0,0)
MAVEC	(0,0)
NAMWRT	(0,0)
SETC	(0,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

LPLT52 (Y,YMAX,ITIME,IYB,NPAR,ID1,ID2,IWGRA,IFN)

Subroutine.
Produces a graphic display. For parameter descriptions, refer to subroutine LPLT51.

ID1,ID2 Unused.

Referenced by	from Overlay
PLT52	(3,0)

External References to
Numerous COMPLOT subroutines.

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

/XPLT52/	SIMCON	(0,0)

LSIZEB (IBUFF,LEN)

Function.
Returns the position of the last non-blank character in a string.

IBUFF	Contains a string in A1 format.
LEN	The length of IBUFF.

Referenced by	from Overlay
LPLT51	(0,0)
REED	(1,0)

External References to	in Overlay
EQC	(0,0)

MACIO (IFN,IBUFF,LEN,IREC,IRET)

Subroutine.
Machine dependent routine which performs I/O functions on the macro library file.

IFN	A function code: 1 Read a record. 2 Write a record. 3 Open a new or existing library file. 4 Close the library file.
IBUFF	An integer array through which the macro text is passed formatted in macro packed form. (See common block /ZZZZ/ for a description.)
LEN	The length of IBUFF.
IRET	Return condition code: 0 Operation complete. 1 I/O error.

Referenced by	from Overlay
GETMAC	(1,1)
PUTMAC	(1,1)
SYSFN	(0,0)

Common Blocks Used	Defined by	in Overlay
/MET/	BLKDAT	(0,0)
/UINDEX/	UCOMIO	(0,0)

MACROS (IARRAY,LEN,IFN,IRET)

Subroutine.
Processes macro definition and execution commands.

IARRAY	A character array containing a command image in A1 format.
LEN	The length of IARRAY.
IFN	A function code: 1 Process a macro call. 2 Write a newly created macro to the library. 3 IARRAY contains a command to be used as part of the macro currently being constructed. 4 Process the macro definition command. 5 Command stack contains a macro; write to library.
IRET	Return condition code: 0 Operation complete. 1 Macro unknown. 2 Macro processing error.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
CHANGE	(1,1)
EQCMP2	(0,0)
FANDC	(0,0)
GETMAC	(1,1)
MAVEC	(0,0)
MOVEW	(0,0)
PACKC	(0,0)
PUTMAC	(0,0)
SCAN	(1,0)
SCSET	(1,0)
SETC	(0,0)
STKMGR	(0,0)
STRFND	(1,1)
UNPACK	(0,0)

Common Blocks Used	Defined by	in Overlay
/MKRF/	BLKDAT	(0,0)
/MPST/	BLKDAT	(0,0)
/NBCM/	BLKDAT	(0,0)
/WKSP/	BLKDAT	(0,0)
/ZZZZ/	BLKDAT	(0,0)

MASTER

Overlay (1,1) Main Program.
This is the SIMCON master controller. The interactive command mode, macro execution, intervention commands (AT commands), model interation and overlay loading are ultimately controlled through this routine. Other overlay main programs including the SIMCON main program serve generally as overlay switching extensions of the MASTER program.

Referenced by	from Overlay
SHAR	(1,0) via CALL OVERLAY

External References to	in Overlay
ATS	(1,1)
CMDFIL	(1,1)
EQC	(0,0)
EQCMP	(0,0)
FILE	(1,1)
GETVAR	(1,1)
GRAPH	(1,1)
HDUMP	(1,1)
INITCM	(1,0)
LOOPER	(1,1)
MACROS	(1,1)
NAMCMD	(1,1)
NAMVAR	(1,1)
PRINT	(1,1)
REED	(1,0)
SCAN	(1,0)
SCSET	(1,0)
SETC	(0,0)
SIMLT	(1,1)
STATS	(1,1)
STKMGR	(0,0)
SYSCMD	(1,1)
SYSFN	(0,0)
SYSTAT	(0,0)
TIMEF	(1,1)

UCMD2	(1,1)
UCMD3	(1,1)
UCMD4	(1,1)
UCMD5	(1,1)
UCMD6	(1,1)
UCMD7	(1,1)
UCOMAN	(1,1)
UINIT	(1,1)
VALFND	(1,0)
VIEW	(1,1)
WHICHC	(1,1)

Common Blocks Used	Defined by	in Overlay
/KCC/	HCOM	(0,0)
/OVTALK/	SIMCON	(0,0)
/SUPCOM/	SIMCON	(0,0)
/WKSP/	BLKDAT	(0,0)
/XPLT52/	SIMCON	(0,0)

MAVEC (N,FSTR,FPOS,TSTR,TPOS)

Subroutine.
Moves an A1 formatted string into another string.

N	The number of characters to move.
FSTR	The starting element of a character string in A1 format.
FPOS	The position of the first character to move from FSTR.
TSTR	The starting element of a character string to move to.
TPOS	The first character position to move to.

Referenced by	from Overlay
DZB	(1,0)
GETMAC	(1,1)
LOOPER	(1,1)
LPLT51	(0,0)
MACROS	(1,1)
NAMVAR	(1,1)
PUTMAC	(1,1)
RDSTMT	(1,2)
RPUTS2	(1,1)
RTAKE2	(1,1)
SCSET	(1,0)
STATS	(1,1)
STGT	(1,1)
STRFND	(1,1)
SUBDET	(1,0)

VARMV	(0,0)
VIEWX	(1,1)

MOVEST (N,ARRAY1,ICHAR1,ARRAY2,ICHAR2)

Subroutine.
A routine analogous to subroutine MAVEC. Moves a packed string or substring into another packed string.

N	The number of characters to move.
ARRAY1	First word address of an integer array containing a packed character string.
ICHAR1	Position of the first character of the substring to be moved counting from the left most position of the first word as position number one. ICHAR1 and ICHAR2 should not be bounded by the number of characters that may fit into an integer, they should be able to take on any positive integer.
ARRAY2	First word address of an integer array to receive the substring.
ICHAR2	Position to place the first character of the substring within ARRAY2.

Referenced by	from Overlay
CHANGE	(1,1)

External References to	in Overlay
GBYTE	(0,0)
PBYTE	(0,0)

MOVEW (NCHAR,ARRAY1,ARRAY2,OLEN)

Subroutine.
Moves full integer words containing packed character strings from one integer array to another given the number of characters to be moved. In situations where substrings begin on word boundaries, this routine is considerably faster than subroutine MOVEST. This routine is system independent.

NCHAR	The number of characters to move.
ARRAY1	An integer array containing a packed character string.
ARRAY2	The first word address of an integer array to move to.
OLEN	The number of full integer words moved returned.

Referenced by	from Overlay
MACROS	(1,1)
PUTMAC	(1,1)

Common Blocks Used	Defined by	in Overlay
/NBCM/	BLKDAT	(0,0)

NAMCMD (IBUFF,LEN)

Subroutine.
Changes the name of an existing SIMCON command.

IBUFF	A character string containing a command image in A1 format.
LEN	The length of IBUFF.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
EQCMP2	(0,0)
PACKC	(0,0)
SCAN	(1,0)
SCSET	(1,0)
SETC	(0,0)

Common Blocks Used	Defined by	in Overlay
/CMDLST/	BLKDAT	(0,0)
/CMDNUM/	BLKDAT	(0,0)
/WKSP/	BLKDAT	(0,0)

NAMVAR (IBUFF,LEN)

Subroutine.
Processes the NAME command.

IBUFF	A command image in A1 format.
LEN	The length of IBUFF.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
FANDC	(1,0)
FANDST	(1,0)
IHSH	(1,0)
MAVEC	(0,0)
SCAN	(1,0)
SCSET	(1,0)
SETC	(0,0)
SUBDET	(1,0)

NAMWRT (NPAR,IWGRA,LOGU)

Subroutine.
Outputs variable names and maximums for printer plots.

NPAR	The number of variables.
IWGRA	A function code: 1 The variables to be output are currently in the plot queue. 2 The variables were named explicitly on the command.
LOGU	The logical unit number to receive the output.

Referenced by	from Overlay
LPLT51	(0,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

PACKC (LEN,CHARS,INTS,OLEN)

Subroutine.
A system dependent character packing routine used to conserve space when storing characters. Characters are packed into integer arrays such that each word holds as many characters as is practical.

LEN	The number of characters to pack.
CHARS	A character string array in A1 format.
INTS	An integer array to receive the characters in packed form.
OLEN	The length in words of array INTS returned.

Referenced by	from Overlay
ATMGR	(0,0)
ENTR	(1,0)
FINDH	(1,0)
GETMAC	(1,1)
MACROS	(1,1)
NAMCMD	(1,1)
PUTMAC	(1,1)
STATS	(1,1)
STKMGR	(0,0)
WHICHC	(1,1)

External References to	in Overlay
PBYTE	(0,0)

PBYTE (BUFF,POS,CHAR)

Subroutine.
A system dependent routine which packs a single character.

BUFF	An integer array into which the character will be placed.
POS	The position at which the character will be placed in BUFF counting as position number one the left most portion of the word. POS may be any positive integer; the proper index into array BUFF is computed.
CHAR	A single character in A1 format.

Referenced by	from Overlay
MOVEST	(1,1)
PACKC	(0,0)

PLT52

Overlay (3,0) Main Program.
Switches between plotting overlay and the command mode overlay.

Referenced by	from Overlay
SIMCON	(0,0) via CALL OVERLAY

External References to	in Overlay
LPLT52	(3,0) via CALL OVERLAY

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)
/XPLT52/	LPLT5	(0,0)

PRINT (INBUFF,LEN,COMON1)

Subroutine.
Processes the PRINT command.

INBUFF	A character array containing a command image in A1 format.
LEN	The length of INBUFF.
COMON1	Unused

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
VIEWX	(1,1)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

PUTMAC (NAME,LN,IBUFF,LEN,IRET)

Subroutine.
Enters a macro into the macro library.

NAME	A macro name in A1 format.
LN	The number of characters in NAME.
IBUFF	An integer array containing the macro text in macro packed form. (See common block /ZZZZ/ for a description of macro packed form.)
LEN	The length of IBUFF in integer words.
IRET	Return condition code: 0 Operation complete. 1 Macro library is full.

Referenced by	from Overlay
MACROS	(1,1)

External References to	in Overlay
EQCMP2	(1,0)
MACIO	(0,0)
MAVEC	(0,0)
MOVEW	(1,1)
PACKC	(0,0)
SETC	(0,0)

Common Blocks Used	Defined by	in Overlay
/MET/	BLKDAT	(0,0)
/WKSP/	BLKDAT	(0,0)

RDSTMT (IBUFF,LAST,LEN,ISTOP,IN)

Subroutine.
Reads a single FORTRAN declaration statement image for symbol table construction. It reads continuation cards up to a total statement length of 660 characters.

IBUFF	A FORTRAN statement image returned.
LAST	The position of the last non-blank character in IBUFF returned. The caller must initialize LAST to 0 before each call.
LEN	The length of the last statement read still in the buffer pending processing. The caller must initialize LEN to 0

before the first statement on the file is processed and never alter it therafter.

ISTOP — Set to one when the end-of-file is reached. The caller must initialize ISTOP to 0 before the first call and never alter it thereafter.

IN — The logical unit number of the input file.

Referenced by	from Overlay
CMREAD	(1,2)

External References to	in Overlay
EQC	(0,0)
MAVEC	(0,0)
REED	(1,0)

REED (IBUFF,LEN,ID1,ID2,LUNT,IRET)

Subroutine.
Reads a card image from the specified file and checks for an end-of-file condition.

IBUFF — A character array containing an image of the line read in A1 format.

LEN — The position of the last non-blank character in IBUFF.

ID1 — Unused.

ID2 — Unused.

LUNT — The logical unit number of the file.

IRET — Return condition code:
1 One line read successfully.
2 The end of the file encountered.

Referenced by	from Overlay
MASTER	(1,1)
RDSTMT	(1,2)

External References to	in Overlay
LSIZEB	(1,0)

RPUTS2 (LIST,LEN,IVAL,OTHER,OTV,ANAME)

Subroutine.
Adds a variable name to the plot queue. The parameters of RPUTS2 make the most sense if they are described in terms the caller's definitions (subroutine LOOPER).

LIST	Equivalenced to the integer array IPLOTQ in common block /SUPCOM/, LIST contains the common block displacements of the variables currently in the plot queue.
LEN	Equivalenced to NPLOT in common /SUPCOM/, contains the number of variables currently in the queue.
IVAL	Contains the common block displacement for the new variable to be entered.
OTHER	Equivalenced to the array PLTMAX in common /SUPCOM/, contains the variable maximums for the variables in the queue.
OTV	Contains the scaling maximum for the variable to be entered.
ANAME	A character array containing the variable name to be entered in A1 format.

Referenced by	from Overlay
LOOPER	(1,1)

External References to	in Overlay
MAVEC	(0,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

RTAKE2 (LIST,LEN,IVAL,OTHER,IRTT)

Subroutine.
Removes a name from the plot queue. See subroutine RPUTS2 for parameter descriptions not listed below.

IRTT	Return condition code: 1 Variable found and deleted. 2 Variable not found.

Referenced by	from Overlay
LOOPER	(1,1)

External References to	in Overlay
MAVEC	(0,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

RTSTUF (ITIME,LENGTH,IATM)

Subroutine.
The model monitoring routine responsible for storing model states between iterations, outputting variables in the plot queue, taking partial sums for run statistics, and causing active AT commands to be executed at the appropriate times.

ITIME	The current simulation time in the model's "years".
LENGTH	Unused.
IATM	A return condition code: 1 Continue model iteration. 2 There are AT commands pending execution.

Referenced by	from Overlay
DOMDL	(2,0)

External References to	in Overlay
ATMGR	(0,0)
LPLT5	(0,0)
STPSUM	(0,0)
UCOMIO	(0,0)
VARMV	(0,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)
/WKSP/	BLKDAT	(0,0)

SCAN (BUFF,LEN,LWDS,NWDS)

Subroutine.

Scans a character string for symbolic substrings or names. A call to SCAN should be preceeded by a call to subroutine SCSET to define delimiter characters.

BUFF	A character string in A1 format.
LEN	The length of BUFF.
LWDS	A doubly subscripted array returned containing the starting and ending positions of each symbolic string found. For example, LWDS(1,1) contains the position of the first character of the first symbol (counted from left to right) in BUFF and LWDS(2,1) contains the position of the last character of the first symbol, etc.
NWDS	The number of symbols found.

Referenced by	from Overlay
ATS	(1,1)
CMREAD	(1,2)
DZB	(1,0)
MACROS	(1,1)
MASTER	(1,1)
NAMCMD	(1,1)
NAMVAR	(1,1)
SIMLT	(1,1)
STGT	(1,1)
SUBDET	(1,0)
TIMEF	(1,1)
VGTS	(1,1)

External References to	in Overlay
FANDC	(1,0)

Common Blocks Used	Defined by	in Overlay
/SCCOM/	SCAN	(1,0)

SCAN2 (BUFF,LEN,LWDS,NWDS)

Subroutine.
Similar to subroutine SCAN except that SCAN2 operates directly on packed strings. Refer to subroutine SCAN for parameters not described below. SCAN2 is system independent.

BUFF	An integer array containing a packed character string.
LEN	The number of characters in BUFF.

Referenced by	from Overlay
CHANGE	(1,1)

External References to	in Overlay
EQC2	(0,0)

Common Blocks Used	Defined by	in Overlay
/SCCOM/	SCAN	(1,0)

SCSET (ID,ND)

Subroutine.
Initializes delimiters in preparation for a call to subroutine SCAN.

ID	A string array of delimiter characters in A1 format.
ND	The number of delimiters.

Referenced by	from Overlay
ATS	(1,1)
CHANGE	(1,1)
CMREAD	(1,2)
DZB	(1,0)
MACROS	(1,1)
MASTER	(1,1)
NAMCMD	(1,1)
NAMVAR	(1,1)
SIMLT	(1,1)
STGT	(1,1)
SUBDET	(1,0)
TIMEF	(1,1)
VGTS	(1,1)

External References to	in Overlay
MAVEC	(0,0)

Common Blocks Used	Defined by	in Overlay
/SCCOM/	SCAN	(1,0)

SETC (NUM,BUFF,CHAR)

Subroutine.
Sets one or more elements of a character string to a common value. It is sometimes also used to zero common blocks or arrays.

NUM	The number of elements to set.
BUFF	The first word address of the array to set.
CHAR	A single character in A1 format.

Referenced by	from Overlay
GETMAC	(1,1)
INITER	(1,2)
LPLT51	(0,0)
MASTER	(1,1)
NAMVAR	(1,1)
PUTMAC	(1,1)
SIMLT	(1,1)
STGT	(1,1)
SUBDET	(1,0)
VIEWX	(1,1)

SHAR

Overlay (1,0) Main Program.
Switches between the initializing overlay and the command mode overlay.

Referenced by	from Overlay
SIMCON	(0,0) via CALL OVERLAY

External References to	in Overlay
INIT	(1,2) via CALL OVERLAY

MASTER	(1,1) via CALL OVERLAY

Common Blocks Used	Defined by	in Overlay
/OVTALK/	SIMCON	(0,0)

SIMCON

SIMCON Main Program, Overlay (0,0)
Prints the SIMCON banner and otherwise only an overlay switching routine.

External References to	in Overlay
DOMDL	(2,0) via CALL OVERLAY
PLT52	(3,0) via CALL OVERLAY
SHAR	(1,0) via CALL OVERLAY

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)
/OVTALK/	SIMCON	(0,0)

SIMLT (INBUFF,LEN,ICMD,IRTT)

Subroutine.
Processes the SIMULATE, CONTINUE, and GO commands, sets simluation parameters, and returns to the caller so that the user's model may be invoked.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
FANDST	(1,0)
FGET	(0,0)
SCAN	(1,0)
SCSET	(1,0)
SETC	(0,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

STATS (INBUFF,LEN,IC)

Subroutine.
Processes the STATS and ONSTAT commands.

INBUFF	A command image in A1 format.
LEN	The length of INBUFF.
IC	The command number of the function to perform. (See the commented table of command numbers listed in program MASTER.)

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
FANDST	(1,0)
MAVEC	(0,0)
PACKC	(0,0)
STGT	(1,1)
STPRT	(0,0)
STPSUM	(0,0)
STZER	(0,0)
UCOMIO	(0,0)
VALFND	(1,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)
/HELP/	STATS	(1,1)
/SIZCOM/	VARMV	(0,0)
/STTCS/	STPSUM	(0,0)

STGT (BUFF,LN,NAME,INL,IDATA,ITAB,IV,IRTT)

Subroutine.
Extracts a name and its subscripts from a command line.

BUFF	A command image in A1 format.
LN	The length of BUFF
NAME	A variable name returned.
INL	The length of NAME.
IDATA	Variable attributes (see common block /COMCOM/ for a description of variable attributes).
ITAB	Subscripts of variable NAME.
IV	Set to zero on the first call by the caller, is incremented by one for each variable processed on the command line.
IRTT	Return condition code: 1 Normal return. 2 End of the line encountered. 2 Syntax error.

Referenced by	from Overlay
LOOPER	(1,1)
STATS	(1,1)
VIEWX	(1,1)

External References to	in Overlay
FANDC	(1,0)
IHSH	(1,0)
MAVEC	(0,0)
SCAN	(1,0)
SCSET	(1,0)
SETC	(0,0)
SUBDET	(1,0)

Common Blocks Used	Defined by	in Overlay
/STGTCM/	STGT	(1,1)

STKMGR (ICMD,IARRAY,LEN,IFN,IRET)

Subroutine.
Handles the details of command stack manipulation.

ICMD	A command image in A1 format.
IARRAY	An integer array used to pass portions of the stack in packed form. (See common block /CSTACK/ for details.)
LEN	The number of characters in string ICMD or the length of IARRAY in integers, whichever is appropriate.
IFN	A function code:

1. Pop the stack, i.e. release the first command from the stack and return it to the caller in ICMD.
2. Push the stack, i.e. place the command contained in ICMD onto the stack.
3. Clear the stack.
4. Set the stack stop point at the current stack pointer. This device makes the stack appear empty under certain conditions when it is not actually empty to cause commands to be preferentially read from a command file or pending AT commands to be executed.
5. Clear the stop point.
6. Enquire if the stack is currently empty. IRET = 0 is returned if it is not empty and IRET = 1 otherwise.
7. Load the stack directly from IARRAY. IARRAY contains a macro ready to execute or one or more AT commands already in packed form.
8. Empty the stack directly into IARRAY.

Referenced by	from Overlay
ATMGR	(0,0)
DOMDL	(2,0)
INIT	(1,2)
MACROS	(1,1)
MASTER	(1,1)

External References to	in Overlay
PACKC	(0,0)
UNPACK	(0,0)

Common Blocks Used	Defined by	in Overlay
/CSTACK/	BLKDAT	(0,0)
/WKSP/	BLKDAT	(0,0)

STPRT

Subroutine.
Prints statistics of simulation variables.

Referenced by	from Overlay
DOMDL	(2,0)
STATS	(1,1)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)
/STTCS/	STPSUM	(0,0)

STPSUM (INT)

Subroutine.
Computes partial sums for simulation variables.

INT Simulation iteration count in the model's years.

Referenced by	from Overlay
RTSTUF	(2,0)
STATS	(1,1)

Common Blocks Used	Defined by	in Overlay
//	CCOM	(0,0)
/STTCS/	STPSUM	(0,0)

STRFND (BUFF,LN,STRING,LEN,STRVAL,LNG,IRTT)

Extracts a key-word assignment from a command line. A key-word assignment has the form "key-word=string".

BUFF	A command image in A1 format.
LN	The length of BUFF.
STRING	The key-word substring for which to search in BUFF.
LEN	The length of STRING.
STRVAL	The character substring to which the key-word is assigned.
LNG	The length of STRVAL.

IRTT Return condition code:
1 Normal return.
2 Key-word not found.
3 Key-word found but there was no assignment.

Referenced by	from Overlay
LOOPER	(1,1)
MACROS	(1,1)

External References to	in Overlay
FANDC	(1,0)
FANDST	(1,0)
MAVEC	(0,0)

STZER

Subroutine.
Initializes the statistical collection process for the variables in the statistics queue.

Referenced by	from Overlay
DOMDL	(2,0)
STATS	(1,1)

Common Blocks Used	Defined by	in Overlay
/STTCS/	STPSUM	(0,0)

SUBDET (ISTR,IL,ITAB,NAME,IRET)

Subroutine.
Interprets subscript ranges for variables specified on command lines.

ISTR A command image in A1 format.
IL The length of ISTR.
ITAB Doubly subscripted array containing subscript ranges for the variable NAME.
NAME A variable name in A1 format.
IRET Return condition code:
1 Variable and subscript ranges returned.
2 Syntax error, not standard FORTRAN.

Referenced by	from Overlay
CONROT	(1,2)
NAMVAR	(1,1)
STGT	(1,1)

External References to	in Overlay
DZB	(1,0)
FANDST	(1,0)
MAVEC	(0,0)
SCAN	(1,0)
SCSET	(1,0)
SETC	(0,0)

SYINIT

Subroutine.
Usually will perform sytem dependent initializations required at the start of a SIMCON run. Often, this routine will open files that are to be used. SYINIT is provided as a dummy routine.

Referenced by	from Overlay
INIT	(1,2)

External References to	in Overlay
MACIO	(0,0)
SYSFN	(0,0)

Common Blocks Used	Defined by	in Overlay
/UINDEX/	UCOMIO	(0,0)

SYSCMD (IBUFF,LEN,IERR,IRET)

Subroutine.
Processes system functions by command. Provided as a dummy routine.

IBUFF	Command image in A1 format.
LEN	The length of IBUFF.
IERR	Error code defined by installation.

IRET Return condition code defined by installation.

Referenced by	from Overlay
MASTER	(1,1)

SYSFN (N)

Subroutine.
Performs system dependent functions at installation option. Many file manipulations would normally be located here such as opening, closing, or rewinding if appropriate. Most entries are provided as dummy but comments in the source may serve as a guide.

N A function code defined by installation.

Referenced by	from Overlay
INIT	(1,2)
MASTER	(1,1)

External References to	in Overlay
MACIO	(0,0)

Common Blocks Used	Defined by	in Overlay
/SYSCOM/	SYSFN	(0,0)
/UINDEX/	UCOMIO	(0,0)

SYSTAT (N)

Logical Function.
Tests system conditions for which a true-false answer is desired.

N Function code:
- 1 Check the attention interrupt flag set by subroutine ATTN (or a user interrupt from an interactive terminal). Return TRUE if set and FALSE if not set.
- 2 Test job origin. Return TRUE if job is timeshare origin, FALSE otherwise.

Referenced by	from Overlay
ATS	(1,1)
DOMDL	(2,0)
HDUMP	(1,1)
LPLT5	(0,0)
LPLT52	(3,0)
MASTER	(1,1)
VIEWX	(1,1)

Common Blocks Used	Defined by	in Overlay
/SYSCOM/	SYSFN	(0,0)

TASET (ITT)

Subroutine.
Merges information about a variable into the symbol table.

ITT — A code for the type declaration statement being processed. See common block /COMCOM/ for type declaration codes.

Referenced by	from Overlay
CONROT	(1,2)

External References to	in Overlay
IHSH	(1,0)

Common Blocks Used	Defined by	in Overlay
/COMCOM/	CMREAD	(1,2)
/SIZCOM/	VARMV	(0,0)
/SUPCOM/	SIMCON	(0,0)

TIMEF (IBUFF,LEN)

Subroutine.
Processes the TIME command.

IBUFF	A command image in A1 format.
LEN	The length of IBUFF.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
FGET	(0,0)
SCAN	(1,0)
SCSET	(1,0)
UCOMIO	(0,0)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

UCOMAN (IBUFF,LEN)

Subroutine.
Optional user provided routine to perform user defined functions by command. By default, a dummy is loaded from the SIMCON library. Also provided but not listed in this guide are six identical routines: UCMD2, UCMD3, UCMD4, UCMD5, UCMD6, and UCMD7.

IBUFF	A command image in A1 format.
LEN	The length of IBUFF.

Referenced by	from Overlay
MASTER	(1,1)

UCOMIO (IFN,LEN,IREC,IRET)

Subroutine.
Processes all I/O functions on the common block dump file.

IFN	A function code: 1 Read one record. 2 Write one record. 3 Read a SAVE record (a record stored for a special purpose apart from other records and usually temporary). 4 Write a SAVE record.
LEN	Record length.
IREC	Record number.
IRET	Return condition code: 1 Operation complete. 2 I/O error.

Referenced by	from Overlay
RTSTUF	(2,0)
STATS	(1,1)
TIMEF	(1,1)
VIEWX	(1,1)
VWGET	(1,1)

Common Blocks Used	Defined by	in Overlay
/UINDEX/	SYSFN	(0,0)
//	CCOM	(0,0)

UINIT

Subroutine.
Optional user supplied routine to perform user defined model initialization tasks. A dummy default is otherwise loaded from the SIMCON library.

Referenced by	from Overlay
INIT	(1,2)

UMODEL (IYEAR)

Subroutine.
The main routine for the user's model.

ITIME The value of the iteration counter defined by SIMCON commands.

Referenced by	from Overlay
DOMDL	(2,0)

Common Blocks Used	Defined by	in Overlay
//	CCOM	(0,0)

UNPACK (LEN,INTS,CHARS,OLEN)

Subroutine.
The logical compliment to subroutine PACKC. See PACKC for parameter defintions.

Referenced by	from Overlay
ATS	(1,1)
HDUMP	(1,1)
MACROS	(1,1)
STKMGR	(1,1)

External References to	in Overlay
GBYTE	(0,0)

VALFND (BUFF,LN,STRING,LEN,VALUE,IRTT)

Subroutine.
Extracts a key-word assignment to a real number in a command line. A key-word assignment has the form "key=number" such as "XVAL=10".

BUFF A command image in A1 format.
LN The length of BUFF.
STRING A key-word in A1 format.
LEN The length of STRING.

VALUE Value returned as a floating point number.
IRTT Return condition code:
1 Key-word assignment extracted.
2 Key-word not found.
3 Syntax error.

Referenced by	from Overlay
LOOPER	(1,1)
MASTER	(1,1)
STATS	(1,1)
VIEWX	(1,1)

External References to	in Overlay
FANDC	(0,0)
FANDST	(1,0)
FGET	(1,0)
IGNORE	(0,0)

VARGET (K,V)

Subroutine.
Picks up the value of a variable from blank common.

K The common block displacement of the value. If K is flagged negative, the value is to be extracted as integer and converted to floating point.
V The value returned in floating point form.

Referenced by	from Overlay
VWGET	(1,1)

External References to	in Overlay
VARMV	(0,0)

VARMV (IFN,ITYP,ICOMON,IDISP,VALUE)

Subroutine.
Extracts and places values in the blank common block and common /SUPCOM/.

IFN	A function code: 1 Get a value from common. 2 Put a value into common.
ITYP	Variable type: 1 Integer 2 Integer * 2 3 Real 4 Logical 5 Logical * 1
ICOMAN	Common block code: 1 SUPCOM 2 Blank
IDISP	Common block displacement in "bytes". See common block /NBCM/ for a definition of "byte".
VALUE	Value returned. VALUE is "typeless" as it is not converted in any way.

Referenced by	from Overlay
LOOPER	(1,1)
RTSTUF	(2,0)
VARGET	(1,1)

External References to	in Overlay
MAVEC	(0,0)

Common Blocks Used	Defined by	in Overlay
//	CCOM	(0,0)
/SIZCOM/	VARMV	(0,0)
/SUPCOM/	SIMCON	/000

VGTS (BUFF,LN,ITYPE,VAL,IENT,IRTT)

Subroutine.
Evaluates the right hand side of a SET command.

BUFF	A command image in A1 format.
LN	The length of BUFF.
ITYPE	Variable type code: 1 Integer 2 Integer * 2 3 Real 4 Logical 5 Logical * 1
VAL	The value returned. VAL is returned in a form according to the variable type.
IENT	Set to zero by the caller on the first call and incrimented by one for each value processed.
IRTT	Return condition code: 1 Value returned. 2 Syntax error.

Referenced by	from Overlay
LOOPER	(1,1)

External References to	in Overlay
FANDC	(1,0)
FANDST	(1,0)
FGET	(1,0)
SCAN	(1,0)
SCSET	(1,0)

Common Blocks Used	Defined by	in Overlay
/VGTSCM/	VGTS	(1,1)

VIEW (IBUFF,LEN,COMON1)

Subroutine.
Processes the VIEW command.

IBUFF	A command image in A1 format.
LEN	The length of IBUFF.
COMON1	Unused.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
VIEWX	(1,1)

Common Blocks Used	Defined by	in Overlay
/SUPCOM/	SIMCON	(0,0)

VIEWX (INBUFF,LEN,IFN)

Subroutine.
Analyzes in detail the VIEW, PRINT, GRAPH, GET, and FILE commands and causes output.

INBUFF	A command image in A1 format.
LEN	The length of INBUFF.
IFN	A function code:
	1 printer plot output (VIEW command).
	2 PRINT command output.
	3 Graphic output (time series).
	4 Graphic output (variable versus variable).
	5 FILE command output.
	6 Get and store variable(s) internally (GET command).

Referenced by	from Overlay
FILE	(1,1)
GETVAR	(1,1)
GRAPH	(1,1)
PRINT	(1,1)
VIEW	(1,1)

External References to	in Overlay
EQC	(0,0)
FANDC	(0,0)
FANDST	(1,0)
LPLT5	(0,0)
MAVEC	(0,0)
PACKC	(0,0)
SETC	(0,0)
STGT	(1,1)

SYSTAT	(0,0)
UCOMIO	(0,0)
VALFND	(1,0)
VWGET	(1,1)

Common Blocks Used	Defined by	in Overlay
/HELP/	VIEWX	(1,1)
/FMTOUT/	VIEWX	(1,1)
/SIZCOM/	VARMV	(0,0)
/SUPCOM/	SIMCON	(0,0)
/WKSP/	BLKDAT	(0,0)

VPRNT (IDATA,NAME,VALUE,I1,I2,I3,I4)

Subroutine.
Prints a variable name and its value for a DISPLAY command.

IDATA	Variable attributes from the symbol table. See common block /COMCOM/ for details.
NAME	A variable name in A1 format.
VALUE	The variable value. The variable type of VALUE is indicated from the attribute type code.
I1,I2,I3,I4	the variable element subscripts provided by the caller if appropriate.

Referenced by	from Overlay
LOOPER	(1,1)

VWGET (P,NV,IGETV,LENGTL,IY,IYMAX,IRET)

Subroutine.
Extracts variable elements from the blank common of a specified iteration.

P	A real array to return variable values.
NV	Number of elements to get.
IGETV	Array containing the common block displacements for each variable element.
LENGTL	The length of the blank common in "bytes". See common block /NBCM/ for the definition of "byte".
IY	Common block record number derived from the current iteration parameters.
IYMAX	The latest record to process.

IRET Return condition code:
1 Normal return.
2 I/O processing error.

Referenced by	from Overlay
VIEWX	(1,1)

External References to	in Overlay
UCOMIO	(0,0)
VARGET	(0,0)

WHICHC (IBUFF,LEN,ICMD)

Subroutine.
Identifies a command and returns its command number.

IBUFF A command image in A1 format.
LEN The length of IBUFF.
ICMD The command number returned.

Referenced by	from Overlay
MASTER	(1,1)

External References to	in Overlay
EQC	(0,0)
EQCMP2	(0,0)
FANDC	(0,0)
PACKC	(0,0)

Common Blocks Used	Defined by	in Overlay
/CMDLST/	BLKDAT	(0,0)
/CMDNUM/	BLKDAT	(0,0)
/WKSP/	BLKDAT	(0,0)

Appendix B: Alphabetical list of common blocks and parameters.

The following are detailed descriptions of most of the important common blocks used in the SIMCON program. The routines and overlays defining each block are given as well as each block element in the order they appear as the block is defined. Array dimensions are given by the number in parentheses following the array name if it is fixed or a star if its dimensions may depend on a particular machine configuration. Individual array elements are specified by the array name followed by the element number enclosed in brackets. Several references are made to "bytes" and "packed" characters. Refer to the description of the common block /NBCM/ for definitions of these terms.

/ATCMDS/ The AT list and associated parameters.
Defined by BLOCK DATA subprogram in overlay (0,0).

IATM (Parameter) The length in full integers for which the array IATLST (below) is dimensioned.

IATN The number of active AT commands.

IATL A pointer to the last element in the list.

IATNDX(40) An integer array containing the index to AT command entries in the list. Each entry is 2 elements in length, element 1 is the effective iteration of the relevant AT command, element 2 is the pointer to the AT command in the list.

IATLST(*) The AT command list, a full integer array dimensioned to a length sufficient hold about 500 packed characters (allowing for 20 AT commands of an average length of about 25 characters) plus 20 integers more. The entries are organized in stack format (see common block /CSTACK/ for a description of stack format).

/CMDLST/ Contains the list of the SIMCON command names and associated parameters.
Defined by BLOCK DATA subprogram in overlay (0,0).

MAX (Parameter) The maximum number of command names for which array space is reserved. 45 is convenient.

IENT (Parameter) The length of a single command name entry in full integers. The length should be sufficient to hold about 20 packed characters.

CMDS(*) An integer array to contain the list of command names. Must be dimensioned to hold no less than the current 34 command name entries but it is recomended that more be reserved to allow command synonyms.

/CMDNUM/ Contains a list of command numbers in one to one correspondence to the names in block /CMDLIST/. See the table of command numbers in the comments in program MASTER.
Defined by BLOCK DATA subprogram in overlay (0,0).

NUMS(*) (Parameter array) Integer array of command numbers (no less than 34 elements, see/CMDLST/).

/COMCOM/ Temporarily stores a symbol table entry description for manipulation by various SIMCON routines.
Defined by subroutine CMREAD in overlay (1,2).

INT(9) An image of a single symbol table entry. Each element is described below:

INT[1] Variable type:
1 Integer
2 Integer * 2
3 Real
4 Logical
5 Logical * 1

INT[2] Displacement from the begining of common in "bytes" starting at 0. See common block /NBCM/ for the definition of "byte".

INT[3] Variable length in bytes.

INT[4] Number of subscripts.

INT[5] Maximum value for left most subscript. If a subscript level does not apply, then that maximum is 1.

INT[6] Maximum value for 2nd subscript.

INT[7] Maximum value for 3rd subscript.

INT[8] Maximum value for 4th subscript.

INT[9] Common block code:
1 Variable element is contained in /SUPCOM/.
2 Variable element is contained in blank common.

/CSTACK/ Contains the command stack and associated parameters. Defined by BLOCK DATA subprogram in overlay (0,0).

IBOT (Parameter) The length in full integers of array ISTACK below.

IBASE A pointer to the base of the stack.

ITOP A pointer to the first entry of the stack.

ISTOP A break-point pointer within the stack.

ISTACK(*) Full integer array containing the command stack. A stack entry is constructed by a "packed" character image of a command preceeded by an integer containing the number of characters in the command. Entries are variable in length but always end on whole integer boundaries. ISTACK should be dimensioned to hold at least 60 twenty-character command strings computed as 60 times the number of integers required to hold 20 packed characters plus 60 integers more. This format is referred elswhere in this guide as "stack format".

/HELP/ Keeps track of variable subscript ranges for processing statistical analyses. Common block elements are defined by EQUIVALENCE to array ITAB calculated in subroutine STGT. Defined by subroutine STATS in overlay (0,0).

ISi Initial value of ith subscript.

IFi Terminal value of ith subscript.

/FMTOUT/ An artifact that is used only by the subroutine (VIEWX) that defines it.

/KCC/ The symbol table and associated parameters. Defined by HCOM subprogram in overlay (0,0).

MAX (Parameter) The length in full integers of array IKDAT below.

IENTSZ (Parameter) The size of an entry in integers. IENTSZ must be 9 plus the number of integers required to hold about 20 packed characters. This parameter is defined in subroutine CONFIG.

NENTS A pointer to the last entry of the table.

IKDAT(*) A full integer array containing the symbol table. It is generally convenient to dimension this array to hold 200 entries of length IENTSZ.

/MET/ The macro entry index table.
Defined by BLOCK DATA subprogram in overlay (0,0).

MMAX (Parameter) The maximum number of entries that may be contained.

MENTSZ (Parameter) The length in integers of a single entry.

MNENTS The number of entries currently held.

MACDAT(*) A full integer array containing the index table. Each entry is comprised of one integer containing the length of a macro in full integers (to pass as a parameter to the random access file routines) and as many integers as is neccesary to hold about 20 packed characters for the macro name.

/MKRF/ A working area to hold the symbolic key-word table of a macro.
Defined by BLOCK DATA subprogram in overlay (0,0).

KSIZE (Parameter) The entry size of a key-word. Simply the number of integers required to hold about 20 packed characters.

KELIST(*) The key-word table. Dimensioned to hold 10 entries.

/MPST/ A working area to hold the macro parameter substitutions for a particular macro call.

ISIZE (Parameter) The entry size of a parameter, the number of integers required to hold about 30 packed characters.

IPLIST(*) The parameter table. Dimensioned to hold 10 entries.

/NBCM/ Defines system dependent machine configuration parameters.
Defined by BLOCK DATA subprogram in overlay (0,0).

NBPI (Parameter) Defined to be the number of "bytes" per full integer element. The term "byte" is taken to mean the smallest addressable unit of storage capable of holding a single character. For a word

addressable machine such as the CDC CYBER, one "byte" is equivalent to one word for this purpose.

NBPR (Parameter) The number of "bytes" per real element.

NCPI (Parameter) The maximum number of characters that may be "packed" into a full integer. For some machines, this may be different than 1 and for the CDC CYBER, NCPI is 10.

/OVTALK/ Overlay Communications.
Defined by subroutine SIMCON in overlay (0,0).

ISWA A switch set to zero to signal execution of the first time SIMCON initialization process, then set to 1 or 2 to control the entry point into the SIMCON command overlay main program, routine MASTER, overlay (1,1).

ISWB When set to 1, the user's model is loaded. When set to 2, the graphic display overlay is loaded.

ISWC Controls the entry point into the initialization overlay main program. When not set (ISWC equals 0), all first time initializations are performed. When set to 1, the RESET function is performed (see the SIMCON user's manual, page 20).

/SCCOM/ Utility common block for subroutine SCAN.
Defined by subroutine SCSET in overlay (1,0).

IDEL(10) A string array containing a list of delimiting characters in A1 format.

NDEL The number of delimiters in IDEL.

/SIZCOM/ System dependent parameters defining the "byte" lengths of various element types. Block is defined in subroutine VARMV, the parameters are defined at execution time in routine INIT, overlay (1,2).

IWDSIZ(8) (Parameter array) Contains the sizes in bytes for the variable types:

IWDSIZ[1] Integer

IWDSIZ[2] Integer * 2

IWDSIZ[3] Real

IWDSIZ[4] Logical

IWDSIZ[5] Logical * 1

IWDSIZ[6] Unused

IWDSIZ[7] Unused

IWDSIZ[8] Unused

/STTCS/ Keeps track of intermediate values necessary for statistical analysis of variables set by STAT or ONSTAT.
Defined by subroutine STPSUM in overlay (0,0) and initialized in subroutine STZER.

NTRVLS Number of observations on which statistics are based.

XMAX(10) Maximum value over interval.

XMIN(100) Minimum value over interval.

XSUM(10) Total value over interval.

XSQSUM(10) Total of variable values squared over interval.

IINT1 Beginning year of analysis.

IINT2 Last year of analysis.

NNAMS(10) Names of variables being analyzed.

IDSPL(10,2) Displacement and type of variables being analyzed.

/SUPCOM/ SIMCON main utility common block.
Defined by subroutine SIMCON in overlay (0,0). Initialized in subroutine INITER.

DEFMAX Default scaling value for variables in the PLOT queue. Default initialized at 100.

IBAUD Data transmission rate to the interactive terminal and used to calculate delay times for graphic output. The default is 1200 BAUD. IBAUD is not relevent for graphic output other than an interactive graphics terminal.

ICHCB Unused.

IDUM1 Flag set to cause the echo of commands back to the terminal. The default is 0.

IDUM2 Unused.

IDUM3 Graphic output device code:
1 A Tektronix terminal.
2 GERBER plotter.
3 GERBER and Tektronix terminal.

IDUM4 Unused.

IDUMP A switch, when set to 1, causes the storing of the blank common block on a random access file between model iterations. The default is 1.

IFAR Unused.

IGETV(10) Common block displacement table for variables to be output.

IGPLT Output display method code for variables in the plotting queue:
1 Line printer plot.
2 Printed table.
3 Graphic display.
4 Not used.
5 Printed output onto alternate file (FILE function).
6 Store values internally (GET function).

ILPLT Unused.

IPLOTQ(10) The common block displacements for all variables in the plot queue.

IYBEG Represents the "year" of the zero'th iteration. IYBEG is initialized at 0 and redefined by the SIMULATE command.

IYEAR The iteration "year" counter. IYEAR is incrimented from IYBEG to IYEND during simulation.

IYEND The "year" at which simulation is to stop. IYEND is initialized at 0 and redefined by any simulation command.

LENGTL The length of the blank common block in "bytes".

LOGUNT The logical unit number to which output is sent. The default is unit 6, the terminal.

LUIN The logical unit number from which to accept command input. Unit 5 is the terminal, unit 1 is the alternate BATCH file.

LUOUT An alternate unit number to send output. Currently, LUOUT is defined to be unit 6, the terminal.

LSUPCM The length of the SIMCON utility common block, /SUPCOM/ in "bytes".

MHCB Unused.

MODTEK The Tektronix graphic device for which graphic output is to be formatted. The default is the Tektronix model 4010. This parameter is not relevent for graphic output to other devices.

NAMES(20,10) A table of the names of the variables in the plot queue. May contain a maximum of 10 names stored as A1 formatted strings.

NKEP The number of output variables currently stored in the output work space.

NLINE The number of print positions to use for a printer plot. The default is 60 printer positions.

NPLOT The number of variables currently in the plot queue.

NYSKIP The interval at which model states are stored (if switch IDUMP is on i.e. set to one) or output presented for variables in the PLOT queue. If NYSKIP = 1 output is presented at every iteration, if NYSKIP = 2 output is presented at every other iteration, etc. The default is every iteration.

PLTMAX(10) A table of the plot scaling maximums for the variables in the plot queue.

OFILE Unused.

OSTAT A switch when on (set to 1), causes statistical monitoring of all variables in the statistics queue and the results presented at the end of the current simulation. The default is off (set to 0).

VMAXS(10) A table of the plot scaling maximums for the variables named on a command.

VNAME(20,10) A table of the names of the output variables named on the last command.

/SYSCOM/ System status utility common.
Defined by subroutine SYSFN in overlay (0,0).

IFLAG Set when attention interrupts occur and periodically tested by routine SYSTAT.

/UINDEX/ Random access file utility common block. This block is relevant only for CDC NOS systems.
Defined by subroutine UCOMIO in overlay (0,0).

INDEX(104) Index array for the common block dump file.

MACNDX(66) Index array for the macro library file.

/VGTSCM/ An artifact that is used only by the subroutine (VGTS) that defines it.

/WKSP/ A general scratch area used mainly for open ended character packing but sometimes for a general working area.
Defined by BLOCK DATA subprogram in overlay (0,0).

IDUM(*) An integer dummy variable dimensioned to about 100 integers or 200 packed characters, whichever is longer.

/XPLT52/ Output utility common block, buffer, and work space.
Defined by subroutine LPLT5 in overlay (0,0).

STORE(10,101) (Real array) Output buffer and internal storage area for up to 10 variable elements for up to 101 saved iterations (including the "zero'th" iteration).

IFPLT Output ready indicator. When set to 1, buffer is ready to output.

NNPAR The number of variable elements represented in the buffer.

IIWGRA Output variable name location code:
1 Variable names are in the plot queue.
2 Variable names are in the list VNAME (located in common block /SUPCOM/.

IIFN Function code:
1 Not used.
2 Not used.
3 Graphics display of variable against simulated time.

4 Graphics display of variable 1 versus variable 2.

/ZZZZ/ A working buffer for macro processing.
Defined by BLOCK DATA subprogram in overlay (0,0).

LIMSIZ (Parameter) A maximum amount of space in integers sufficient to hold an entire macro. It should be the same length as the command stack array (see common block /CSTACK/).

IBUFF(*) An integer array dimensioned to LIMSIZ plus enough space for 200 or more packed characters.