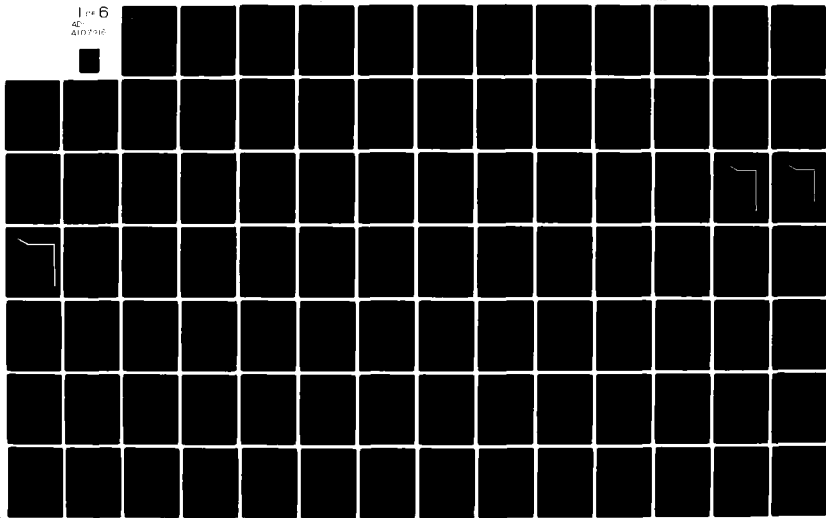


AD-A107 916

BDM CORP MCLEAN VA F/S 15/3
MODULAR AIR DEFENSE EFFECTIVENESS MODEL, PROGRAM DOCUMENTATION --ETC(U)
JAN 80 M FILTEAU, B MACALEER, J T HAWKINS DNA001-79-C-0230
UNCLASSIFIED BDM/W-79-646-TR-VOL-2 DNA-5594F-2 NL

1 of 6
40
A107916



LEVEL III

2

DNA 5594F-2

AD A107916

MODULAR AIR DEFENSE EFFECTIVENESS MODEL, PROGRAM DOCUMENTATION AND USER'S GUIDE

Volume II—MADEM Programmer Manual

The BDM Corporation
7915 Jones Branch Drive
McLean, Virginia 22102

31 January 1980

Final Report for Period 1 March 1979—31 January 1980

CONTRACT No. DNA 001-79-C-0230

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

THIS WORK SPONSORED BY THE DEFENSE NUCLEAR AGENCY
UNDER RDT&E RMSS CODE B325079464 V99QAXNI06215 H2590D.

Prepared for
Director
DEFENSE NUCLEAR AGENCY
Washington, D. C. 20305

DTIC
ELECTE
S DEC 1 1981 D
D

81 11 30 050

Destroy this report when it is no longer
needed. Do not return to sender.

PLEASE NOTIFY THE DEFENSE NUCLEAR AGENCY,
ATTN: STTI, WASHINGTON, D.C. 20305, IF
YOUR ADDRESS IS INCORRECT, IF YOU WISH TO
BE DELETED FROM THE DISTRIBUTION LIST, OR
IF THE ADDRESSEE IS NO LONGER EMPLOYED BY
YOUR ORGANIZATION.



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

12-5-1

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1 REPORT NUMBER DNA 5594F-2	2 GOVT ACCESSION NO. AD A09916	3 RECIPIENT'S CATALOG NUMBER
4 TITLE (and Subtitle) MODULAR AIR DEFENSE EFFECTIVENESS MODEL, PROGRAM DOCUMENTATION AND USER'S GUIDE Volume II—MADEM Programmer Manual	5 TYPE OF REPORT & PERIOD COVERED Final Report for Period 1 Mar 79—31 Jan 80	
	6 PERFORMING ORG. REPORT NUMBER BDM/W-79-646-TR	
7 AUTHOR(S) M. Filteau J. T. Hawkins L. Elfes B. Macaleer J. Philips	8 CONTRACT OR GRANT NUMBER(S) DNA 001-79-C-0230	
	10 PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Subtask V99QAXNI062-15	
9 PERFORMING ORGANIZATION NAME AND ADDRESS The BDM Corporation 7915 Jones Branch Drive McLean, Virginia 22102	12 REPORT DATE 31 January 1980	
	13 NUMBER OF PAGES 534	
11 CONTROLLING OFFICE NAME AND ADDRESS Director Defense Nuclear Agency Washington, D.C. 20305	15 SECURITY CLASS (of this report) UNCLASSIFIED	
	15a DECLASSIFICATION DOWNGRADING SCHEDULE	
14 MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		
16 DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17 DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18 SUPPLEMENTARY NOTES This work sponsored by the Defense Nuclear Agency under RDT&E RMSS Code B325079464 V99QAXNI06215 H2590D.		
19 KEY WORDS (Continue on reverse side if necessary and identify by block number) Modeling Simulation Air Defense Models Discrete Event Simulation		
20 ABSTRACT (Continue on reverse side if necessary and identify by block number) This document is one of three volumes which provide documentation and user guidelines for the Modular Air Defense Effectiveness Model (MADEM). The three volumes are: VOLUME I—MADEM Analyst Manual; VOLUME II—MADEM Programmer Manual; and VOLUME III—MADEM Enhancement Specifications.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

PREFACE

The purpose of this manual is to document the Modular Air Defense Model (MADEM) and its implementation. The manual discusses the software architecture, data structures and execution requirements in detail.

The manual is intended for use by programmers charged with maintaining or modifying MADEM. The MADEM Analyst Manual discusses the processes modeled, their structure and relationships, and the various assumptions made.

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By _____	
Distribution/ _____	
Availability Codes	
Dist	Avail and/or Special
A	

DTIC
SELECTED
DEC 1 1981
S D
D

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
	PREFACE	1
	LIST OF ILLUSTRATIONS	5
I	INTRODUCTION	7
II	SOFTWARE ARCHITECTURE	8
	A. Introduction	8
	B. Implementation Language	8
	C. Data Structures	9
	D. Simulation Control	9
	1. Top Level Control	9
	2. Initialization	13
	3. Storage Management	13
	4. Event Control	14
	E. Diagnostics	17
	1. Debug Routines	17
	2. Data Structure Display Routines	19
	F. Software Components	20
	1. Processor Configuration	20
	2. Preprocessor Functions and Subroutines	22
	3. Main Processor Functions and Subroutines	26
	4. Postprocessor Functions and Subroutines	34
III	DATA STRUCTURE DOCUMENTATION	35
	A. Introduction	35
	B. Common Blocks	40
	C. Data Base Structure-DATFILE	47
	1. Data Block Index	47
	2. Description	47
	3. Structure Overview	48
	4. Block Specifications	50
	5. Linkages to Other Data Structures	59
	6. Notes	60

TABLE OF CONTENTS (CONTINUED)

<u>Section</u>	<u>Page</u>
III	
D. Simulation Control Structures	61
1. Event Lists	62
2. Hex Address Tree and Related Lists	66
E. Command/Control Structures	71
1. Player Lists	72
2. C2 Trees	78
3. Passive Target List	86
F. RED Structures	89
1. Red Theater Commander	90
2. Potential Target List	98
3. Assigned Target List	103
4. Nonavailable Target List	109
5. Corridor Description List	114
6. Attack Requirements/Allocations Lists	121
7. Red Air Bases	130
8. Red Flights	137
G. Blue Structures	153
1. Allied Tactical Air Force (ATAF)	154
2. Sector Operations Centers (SOC)	159
3. Combat Reporting Centers (CRC)	164
4. Blue Airbases (B-AB)	175
5. Blue Flights (B-FLT)	182
6. Battalion Operations Centers (BOC)	198
7. Sam Battery (BTRY)	215
8. Passive Targets (PT)	231
H. Storage Space Management	235
1. Data Block Index	235
2. Description	235
3. Structure Overview	235
4. Block Specifications	238
5. Linkages to Other Data Structures	238
6. Notes	238

TABLE OF CONTENTS (CONTINUED)

<u>Appendices</u>		<u>Page</u>
APPENDIX A	Executive Requirements	239
APPENDIX B	Modular Information Data Access System (MIDAS)	263
APPENDIX C	The Program Design Language	281
APPENDIX D	MIDAS Tables	291
APPENDIX E	The Dynamic event scheduling Algorithm	303
APPENDIX F	MADEM Event Code Definitions	311
APPENDIX G	Random Number Generator Calls	317
APPENDIX H	MADEM Subroutine Reference Lists and Calling Hierarchies	319
APPENDIX I	MADEM Debug Routines	469
APPENDIX J	MADEM Data Structure Cross Reference	481

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
II-1	Top Level Subroutine Calling Hierarchy	10
II-2	Event Control Subroutine Calling Hierarchies	15
II-3	Event Processing Overview	16
II-4	MADEM Processor Configuration	21
III-1	Data Base Structure Diagram	49
III-2	Discreet Event List Diagram	63
III-3	Hex Tree Structure Diagram	67
III-4	Player List Structure Diagram	73
III-5	C2 Tree Structure Diagram	79
III-6	Blue Command/Control Structure	84
III-7	Red Command/Control Structure	85
III-8	Passive Target List Structure Diagram	87
III-9	Red Theater Commander Structure Diagram	91
III-10	Potential Target List Structure Diagram	99
III-11	Potential/Assigned Target List Structures	102
III-12	Assigned Target List Structure Diagram (Red Commander Perception List)	104
III-13	Nonavailable Target List Structure Diagram	110
III-14	Attack Sectors	113
III-15	Corridor Description List Structure Diagram	115
III-16	Relationship of HEX LINK Lists to Corridor Boundaries	120
III-17	Attack Requirements/Allocations List Structure Diagram	123

LIST OF ILLUSTRATIONS (CONTINUED)

<u>Figure</u>	<u>Page</u>	
III-18	Red Air Base Structure Diagram	131
III-19	Red Flight Structure Diagram	139
III-20	Flight Data Base Structure Diagram	140
III-21	Allied Tactical Air Force Structure Diagram	154
III-22	Sector Operations Center Structure Diagram	159
III-23	Combat Reporting Center Structure Diagram	165
III-24	Blue Airbase Structure Diagram	176
III-25	Blue Flight Structure Diagram	184
III-26	Flight Data Base Structure Diagram	185
III-27	Battalion Operations Center Structure Diagram	200
III-28	Digested Information List Configurations	201
III-29	Digested Information List and Perception List	201
III-30	SAM Battery Structure Diagram (HAWK and HERC)	217
III-31	SAM Battery Structure Diagram (PATRIOT)	217
III-32	BTRYDIL Configurations	218
III-32	Battery BTRYDIL Perception List Association	219
III-34	Passive Target List Structure Diagram	232
III-35	I-Space Array Configuration	236
III-36	Garbage Collection Matrix Structure Diagram	237

CHAPTER I
INTRODUCTION

The purpose of this manual is to document the MADEM Software Architecture. It is designed for use by programmers charged with maintaining or modifying MADEM. Those wishing to use MADEM in a study are referred to the MADEM ANALYST MANUAL.

Chapter II of this manual provides an overview of the MADEM Software Architecture. Its primary objective is to associate particular sub-routines with particular simulation control and modeling functions. A secondary objective is to provide a basic explanation of the data storage system used in MADEM. A general knowledge of the subroutines combined with an understanding of the data storage system provide a context within which the more detailed information in Chapter III, the Appendices and the Source Code may be used.

Chapter III of this manual contains detailed Data Structure Documentation. This information is crucial to an understanding of the MADEM Software. MADEM uses a complex list processing system to store and retrieve data. In this system the relationship among various blocks of data is as important to the functioning of the software as the contents of the blocks. Groups of related data blocks form data structures which, in effect, "drive" the software. Therefore, it is impossible to understand the Source Code without a clear picture of the data structures and their contents.

CHAPTER II SOFTWARE ARCHITECTURE

A. INTRODUCTION

The primary objective of this chapter is to associate particular subroutines with the model functions they control. A secondary objective is to introduce the user to the data storage system used in MADEM. More detailed information on subroutines and data structures is contained in the Appendices, the Source Code and in Chapter III of this report.

B. IMPLEMENTATION LANGUAGE

MADEM was implemented in accordance with principles of topdown structured programming. Prior to code generation, the MADEM design was specified in a BDM developed Program Design Language (PDL). The concepts and procedures involved in the use of PDL are discussed in Appendix C. In essence, the PDL for a given subroutine constitutes the equivalent of a logic flow diagram for the subroutine. The PDL for each MADEM subroutine appears in the source code. Information on the basic purpose, the inputs and outputs, and the calling interfaces for each subroutine is given in Appendix I. The overall model structure is reflected by the subroutine call diagrams in Appendix J.

The design specified by the MADEM PDL was implemented in FORTRAN. A special precompiler was used to allow convenient access to MADEM's many dynamically allocated data blocks. This precompiler is the BDM developed Modular Information Data Access system (MIDAS), which is described in Appendix B. MIDAS provides two important capabilities. The first allows automatic replacement of specified one line macro-instructions by corresponding sets of FORTRAN instructions. This feature is used to insure uniformity in the definition of named COMMONs from routine to routine. The second important MIDAS capability allows reference by name to elements of dynamically allocated data blocks. Thus, for example, MIDAS may permit a

reference of the form \$P.TYPE.RATE\$ to be used in place of a corresponding FORTRAN reference ITR(ITR(P+L)+5), where P has been declared a pointer to a data block of a type known to MIDAS.

C. DATA STRUCTURES

Most of the data employed by MADEM is stored internally in data blocks dynamically allocated from various storage arrays. Many types of such blocks are employed, each having a block name and most having a set of element names (and types) known to the MIDAS translator. Having established a pointer to such a block and having indicated the block type in a MIDAS "DECLARE" statement, a programmer may then reference any entry in the block by name. Chapter III presents descriptions for all of the MADEM data block types; the MIDAS name for each block is given, along with comments on the use of the block within the simulation. The name, type, and meaning (use) of each element of the block is also given. The structure and contents of each type of data block as implemented in MIDAS code is also indicated in Appendix E.

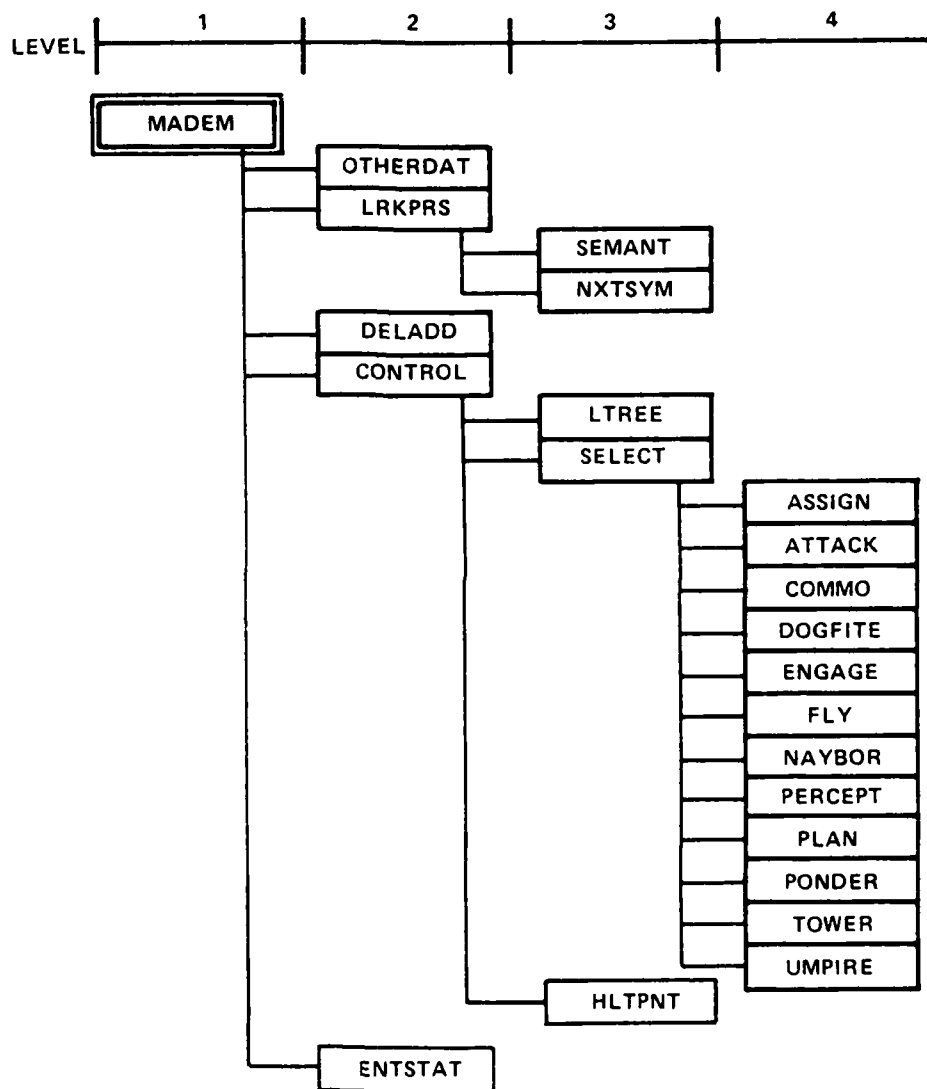
In addition to common storage arrays for dynamically allocated data blocks, MADEM employs a number of other named COMMON storage areas for holding simulation control information and temporary working data. Chapter III presents each of the COMMONs and indicates the meaning (use) of each element.

D. SIMULATION CONTROL

The purpose of this section is to discuss Simulation Control and Management Functions within MADEM and to associate particular subroutines with these functions.

1. Top Level Control

The Top Level Control routines for MADEM are shown in Figure II-1. MADEM is the main routine which invokes the four major control routines; OTHERDAT, LRKPRS, DELADD, and CONTROL. At the users option MADEM



NOTE: ONLY SUBROUTINES WHOSE PRIMARY FUNCTION IS CONTROL HAVE BEEN INCLUDED IN THIS DIAGRAM. FOR A COMPLETE CALLING HIERARCHY SEE APPENDIX J.

may also invoke ENTSTAT which provides a variety of subroutine diagnostics at various points in program execution.

OTHERDAT controls input to the (NON-UOIL) Data Base. The contents and structure of this data base are described in detail in Chapter III of this report. The data input procedure is documented in Chapter IV of the MADEM ANALYST MANUAL.

LRKPRS controls semantic processing of the User Oriented Input Language (UOIL) inputs. The data structures which are built from these inputs are discussed in Chapter III of this report. The Input Language and data requirements are documented in Chapter IV of the MADEM Analyst Manual. LRKPRS invokes two secondary control subroutines- SEMANT and NXTSYM. LRKPRS and NXTSYM parse the input sentences and convert them to "ICODES" which are passed to SEMANT. SEMANT converts these "ICODES" to data structures of various kinds. Users are cautioned to avoid modifications to LRKPRS, NXTSYM and SEMANT unless they are well versed in semantic processing techniques.

DELADD controls processing of discrete events in MADEM. It adds "events" to a Leftist EVENT TREE structure which is used to sort events scheduled by the various program modules. Events control is discussed in further detail in section of D.4 and in Appendix E. of this manual.

CONTROL invokes three major control subroutines- LTREE, SELECT, and HLPNT. LTREE removes "EVENTS" from the EVENT TREE constructed by DELADD. Events are removed and processed in order of their occurrence. These "EVENTS" are then processed by SELECT. Based on an internal coding system for EVENTS, (see Appendix F) SELECT invokes one of nine program modules. Each of these modules (represented in Figure D-1 by their main control routines) performs all of the functions required by the EVENTS. The functions of each module are summarized in Table II-1. Additional information on each module is provided in Section F.3, and Appendix I of this Manual as well as in the source code. If a Termination of Simulation event is found by LTREE, the simulation termination subroutine HLPNT is invoked by CONTROL. HLPNT controls printing of termination messages and output files to be stored for subsequent use by the main processor and post processor.

TABLE II-1.
MADDEM MODULE SUMMARY

<u>MODULE CONTROL/ROUTINE</u>	<u>FUNCTION</u>
ASSIGN	- COMBAT REPORTING CENTER MAKES OF INTERCEPTORS TO RED FLIGHTS
ATTACK	- CARRY OUT GROUP ATTACK BY RED FLIGHTS
COMMO	- COMMUNICATIONS TRANSMISSION
DOGFITE	- AIR TO AIR COMBAT PROCESSES
ENGAGE	- SURFACE TO AIR MISSLE ENGAGEMENT DECISIONS
FLY	- AIRCRAFT MOVEMENT
NAYBOR	- DETERMINE NEARBY UNITS AND SCHEDULE ALL UNITS FOR A CHANCE TO "SEE" AN ACTION
PERCEPT	- CONTROLS PERCEPTION OF OTHER UNITS
PLAN	- RED THREAT PLANNING
PONDER	- UNIT INFORMATION PROCESSING
TOWER	- AIRBASE OPERATIONS
UMPIRE	- SIMULTAINEOUS EVENT CONTROL

2. Initialization

The subroutines devoted solely to initialization are:

- (1) BDALT
- (2) BDLEX
- (3) BDLRK
- (4) BDPARS
- (5) BLKDAT
- (6) FETCH

All but one of these (BLKDAT) are used only by the Lexical Analyzer Routines. They contain parsing tables which are essential to the UOIL decoding process. BLKDAT is used to initialize the MADEM main routine. FETCH reads input data files for the main routine.

3. Storage Management

To provide maximum flexibility in the types of scenarios which can be handled by MADEM, a system of dynamic storage allocation is employed. The majority of data in MADEM is stored in a single array (ISPACE, or SPACE) from which storage space is dynamically allocated. Thus, deletion of data of one type frees space for data of other types. The subroutines exclusively devoted in storage allocation are:

- (1) GIMME
- (2) RELEASE
- (3) RELIST

subroutines GIMME and RELEASE are the general purpose storage management routines employed in MADEM. To obtain a data block of length N ($N=1-20$) the statement.

Call GIMME (P,N) is used. The return value P is a pointer to the allocated block (address of the first word of the block in the ISPACE array). When the block is no longer needed, the statement.

Call RELEASE (P,N) releases the block for subsequent reallocation. The released block is placed in a garbage collection matrix (see Chapter III .H) which consists of lists of blocks of various sizes. When a block of a given size is required, GIMME searches this list to find a release block of the desired size before it allocates virgin storage space.

Because very large blocks are seldom called for, released blocks over 20 words long are broken down into more commonly used four word blocks by the subroutine RELIST. Further details on storage space management are contained in Chapter III of this manual.

4. Event Control

Since MADEM is an event stepped simulation, the management of notices for pending events is an important component of simulation control. The following subroutines are devoted to this function:

- (1) DELADD
- (2) LTREE

DELADD is used to add events to a quisisorted LEFTIST EVENT TREE in which the nearest events in time are placed closest to the top of the tree. The top event block on the tree always contains the next event code to be processed. LTREE is used to extract a pending event from the EVENT TREE. The calling hierachies of DELADD and LTREE are illustrated in Figure II-2.

DELADD invokes three subroutines - GIMME, SNAP, AND LTRMRG. GIMME allocates storage space for the new event block, SNAP adds the newly created block to the event tree, and LTRMRG sorts the event tree to place all of the event blocks in the correct order with respect to the top of the tree.

LTREE removes the event block from the tree, passes the event code stored in the block to SELECT and releases the storage space formerly used by the event block by invoking the subroutine RELEASE.

An overview of event processing is shown in Figure II-3. Event blocks are added to the event tree as actions are taken by the various program modules invoked by SELECT. SELECT is in turn driven by event codes extracted from the event tree by LTREE. This process continues throughout the simulation until a termination event is found by LTREE. When this occurs, HLTPNT is called by CONTROL and the simulation is HALTED. Detailed documentation of this dynamic event scheduling algorithm as well as a listing of event codes may be found in Appendices E and F of this manual.

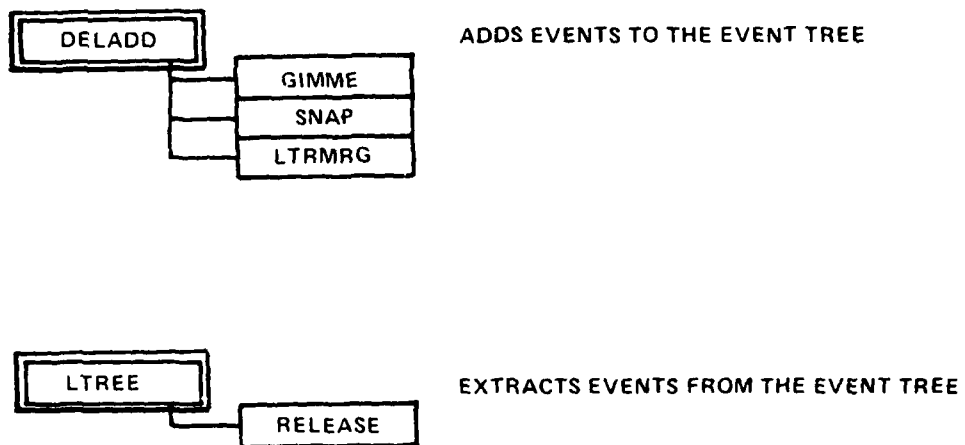
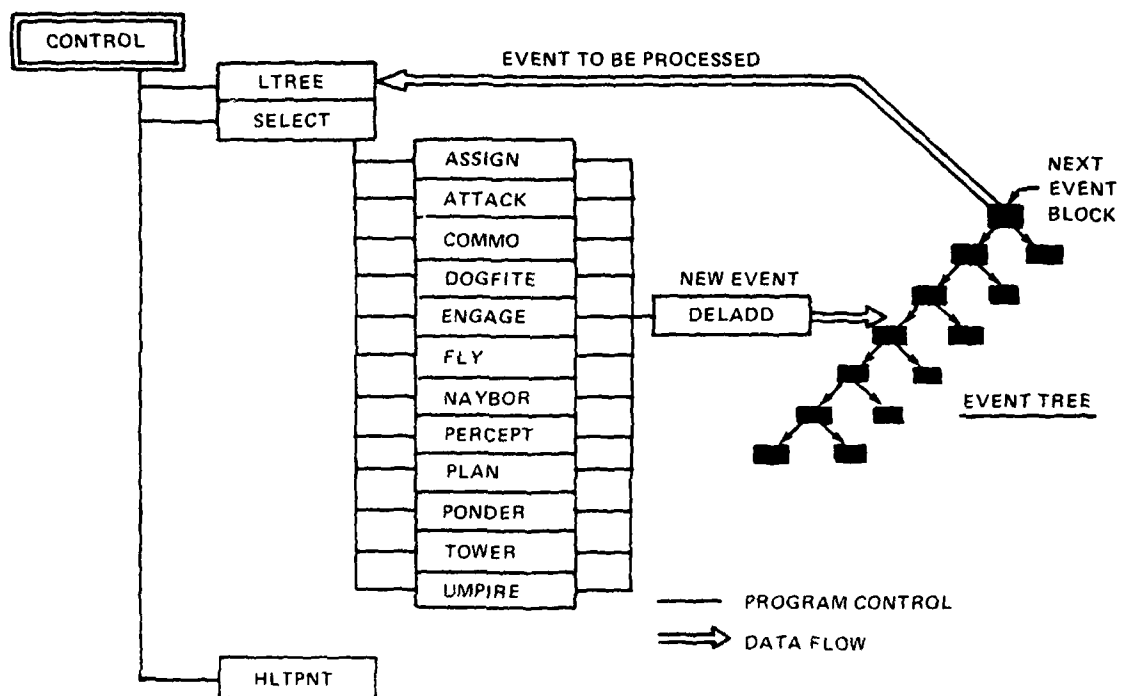


Figure II-2. Event Control Subroutine Calling Hierarchies



4368/79W

Figure II-3. Event Processing Overview

E. DIAGNOSTICS

The purpose of this section is to document the diagnostic capabilities which have been built into the MADEM software. A working knowledge of these debug and data structure display routines is essential to anyone who must maintain or modify MADEM. The operation of all diagnosis subroutines is discussed in detail in Appendix J of this report.

1. Debug Routines

There are over 300 subroutines in the current version of MADEM. A program of this size cannot be developed and maintained without some resident debug and error recovery capability built into the software. In MADEM the following subroutines are devoted to this function:

<u>ENTRYP</u>	<u>DBGREAD</u>
<u>EXITP</u>	<u>HALT</u>
<u>ENTSTAT</u>	<u>ICHECK</u>
<u>RECOVR</u>	<u>ITRAP</u>
<u>RECCON</u>	
<u>RECER</u>	
<u>ROUTER</u>	

MADEM's resident debug capabilities are based on the subroutine entry and exit tracking routines ENTRYP and EXITP. ENTRYP and EXITP are called at the beginning and end of nearly all routines in MADEM. Together they construct a circular list and pushdown stack of subroutine calls. They also construct a vector of subroutine entry counts and a corresponding vector of cumulative subroutine execution times. The circular list and pushdown stack allow the user to track the execution of all subroutines and to pinpoint fatal error locations. The vectors of entry counts and execution times provides useful information on re usage and efficiency of subroutines. At the user's option ENTRYP and EXITP will print out subroutine entry and exit messages for specified subroutines during normal program execution. Also at the user's option, ENTRYP and EXITP will call debug routines ICHECK and ITRAP (explained below) for specified subroutine calls to ENTRYP and EXITP.

A frequent list processing bug involves a zero or one digit ISPACE pointer. For this reason, the first ten words of ISPACE are reserved for nonuse, and should always be zero when none of these bugs occur. Debug routine ITRAP checks that area of ISPACE and stops the simulation by calling HALT whenever nonzero values are found there.

If it is desired to find out when other locations in ISPACE change value, debug routine ICHECK can be used. ICHECK holds pointers to up to ten ISPACE locations, as defined by input parameters on each run. Any time one of these ISPACE locations has a change in value, a one line message is printed indicating the old value, the new value, and the routine being executed when the change was noticed.

The vectors of subroutine entry counts, cumulative and average execution times, and flags indicating the ENTRYIP and EXITIP options specified are printed by ENTSTAT.

RECOR is a CDC System routine which is automatically activated within MADEM, but may be turned off at the user's option. It allows the program to regain control at the time that abnormal job termination would otherwise occur. It calls the MADEM subroutine RECCON in the event of catastrophic program failure.

RECCON can only be activated by RECOR. RECCON calls HALT, which stops the simulation and prints debug information.

ROUTER, which can be called from any routine, prints the subroutine name, the circular list subroutines called, and the pushdown stack containing the calling hierarchy.

RECER is the same as router except that RECER does not know the calling subroutine's name.

DBGREAD is the subroutine that reads, interprets, and processes all the debug input parameters that may be specified for each run. See Appendix J for further details.

HALT is the routine that is called whenever the simulation is to be stopped, whether it be abnormal or normal terminations. HALT saves the HOLD files and calls the appropriate subroutines to print valuable debug information. HALT prints the name of the 1st routine executed as well as

the reason for termination, the game time of the simulation, and the number of events executed in the volume. HALT then calls the following routines:

TRACE - prints current calling hierarchy
RECER - (see above)
ENTSTAT - (see above)
CLIST - prints key pointer values
ISDUMP - if selected, prints entire ISPACE array.

2. Data Structure Display Routines

Because of the complex nature of the data structures used in MADEM and their importance to the operation of MADEM subroutines, a series of data structure and common block display subroutines have been developed. These subroutines include the following:

CLIST
ISDUMP
DISPOAT
NIPULATOR

CLIST is used to print out the contents of all major common blocks used in MADEM. Key information contained in these blocks includes array dimensions and pointer values. ISDUMP is used to print out the contents of the dynamic storage array ISPACE within which all data structures reside. The outputs from CLIST and ISDUMP may be used to manually trace data structures in ISPACE.

However, since manual data structure tracing rapidly becomes time consuming and error prone, a variety of automated data structure display subroutines have been constructed. The two foremost of these are DISPDAT and NIPULATOR. DISPDAT outputs a formatted display of the DATA BASE STRUCTURE, the EVENT TREE STRUCTURE and specified portions of ISPACE.

In addition to the above display subroutines, a series of data block display subroutines has been developed which allows the user to build "custom" data structure display systems to suit special needs. Each data block has its own display subroutine which can display the block or all blocks in a linked list at the user's option. These block display subroutines can be placed under the control of special purpose control routines

developed by the user. User supplied control subroutines combined with the block display subroutines constitute a "modular" data structure display system which can be reconfigured at the user's option. Further details are provided in Appendix J.

F. SOFTWARE COMPONENTS

The purpose of this section is to introduce the reader to the major components of the MADEM software system. Detailed operating instructions may be found in Appendix A of this manual and Chapter IV of the analyst manual.

1. Processor Configuration

MADEM has three major software components - a preprocessor, a main processor, and a postprocessor. The overall configuration of these processors and their input and output files is illustrated in Figure II-4.

The preprocessor (sometimes referred to as INITBIN) reads user inputs, which include the data base and red threat planning specifications, translates these inputs into appropriate data structures and carries out the red threat planning process. The resulting red attack plan is then saved on a "HOLD FILE" for subsequent use by the main processor also outputs a printed summary of the red attack plan and a "HISTORY FILE". The HISTORY FILE contains a record of unit creation events which can be used by the postprocessor to construct summary tables of the types of units created.

The main processor (sometimes referred to as RUNBIN) is used to simulate the actual combat processes which result from the red attack. The main processor is run in a cyclical fashion. The initial main processor run is made using the RED ATTACK PLAN HOLD FILE as input. Combat processes are then carried out for approximately one hour of game time after which execution is terminated and as HOLD FILE containing the status of all units at the end of the hour is output. This HOLD FILE then becomes the input for the next main processor cycle. Each of these main processor cycles is referred to as a volume. As many volumes can be run as one required to simulate the desired duration of conflict. However, the validity of the

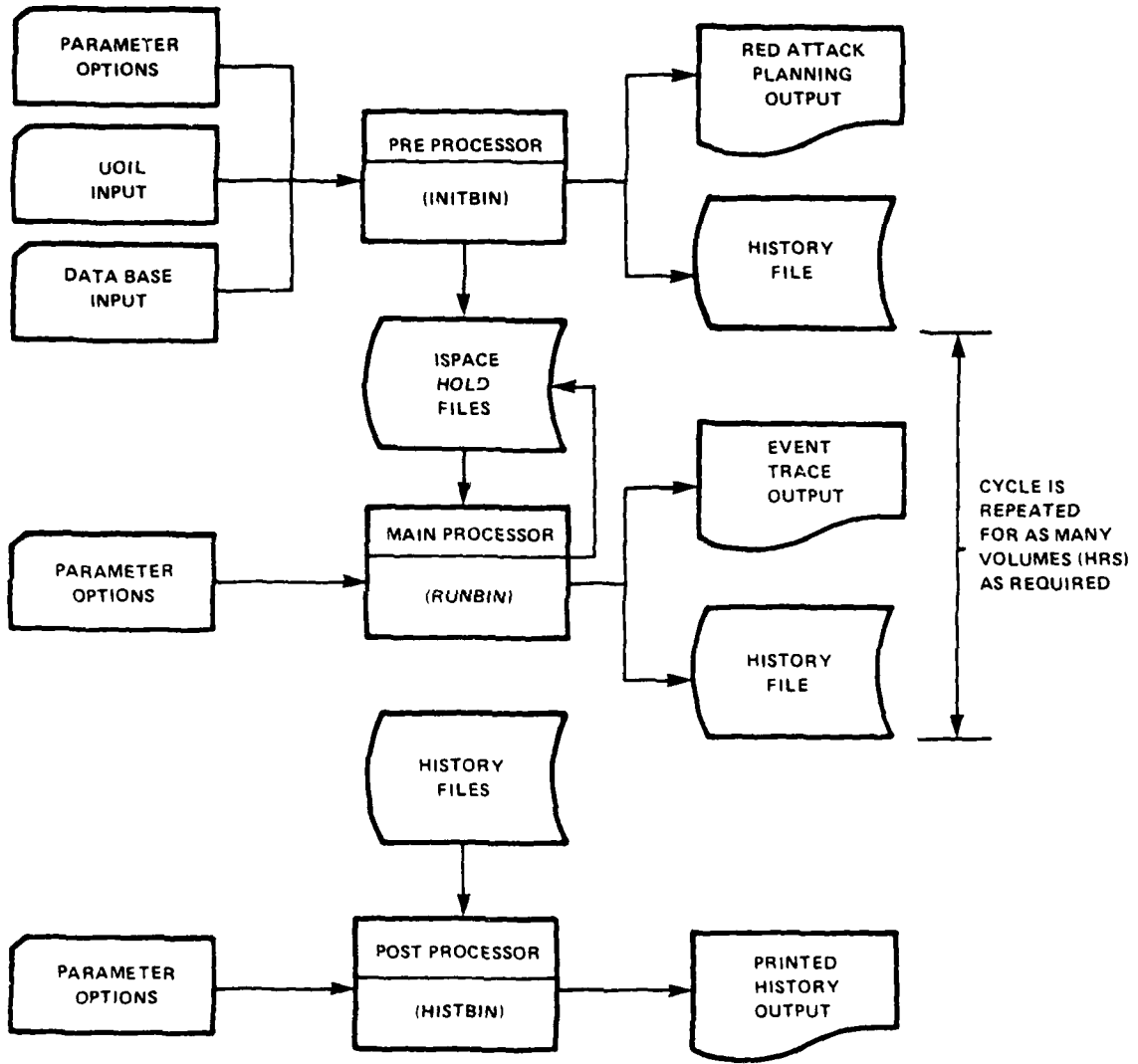


Figure II-4. MADEM Processor Configuration

simulation probably declines rapidly after 8-11 hours of combat. The main processor also outputs an EVENT TRACE and a History File. The EVENT trace records all of combat events which occurred during the volume. The History File contains a record of key events which can be used by the postprocessor to summarize the battle.

The postprocessor (sometimes referred to as HISTBIN) reads History Files output by the preprocessor and main processor and outputs a variety of tabular summaries of the battle. These outputs include the following:

- (1) Red Aircraft acquired by Blue Defense Units
- (2) Red Aircraft engaged by Blue Defense Units
- (3) Red Aircraft damaged by Blue Defense Units
- (4) Blue Units damaged by Red Aircraft
- (5) Weapon System expenditures by Unit Type
- (6) Number of Red Aircraft to reach Targets
- (7) Number of Units created by Type

2. Preprocessor Functions And Subroutines

The following lists summarize the functions carried out by various preprocessor subroutines. Low level utilities have been deleted from these lists. For a complete calling hierarchy and detailed subroutine documentation see Appendices I and II, and the Source Code.

a. Top Level Control Routines

MADDEM	-	Main Routine
OTHERDAT	-	Constructs Simulation Data Base
LRKPRS	-	LR(K) Parser
SEMANT	-	Semantic Processing of UOIL
CONTROL	-	Event Control
SELECT	-	Selects Appropriate Event Module
PLAN	-	Red Threat Planning
HLTPNT	-	Halts Simulation
ENTSAT	-	Simulation software Use Statistics

b. Simulation Data Base Construction

- OTHERDAT - Controls Construction of Data Base
- ADDBLOK - Adds a Data Block to the Top of a Singly
Linked List Structure
- FINDBLOK - Finds a Data Block with a Specified Value in
a Singly Linked List Structure

c. Semantic Processing Of UOIL

- SEMANT - Controls Semantic Processing
- CODE01 - Initialized Hex and C2 Trees
- CODE03 - Constructs C2 Tree and Related Structures
- SRCHPL - Searches the Players List
- LOADPL - Enters a Unit onto the Players List
- ABQUEVE - Sets up Airbase Data Structures
- ADASASS - Constructs CRC, BOC and BTRY Structures
- SCHTAB - Sets Hex Altitude From Altitude Data Base
- CODE05 - Constructs Hex Tree and Related Lists
- VOLLOAD - Places a unit on a Hex's Unit Occupancy List
- TGTLIST - Constructs Target List
- INITACQ - Initializes Acquisition Devices
- CODE18 - Locates Targets in Hex Tree

d. Lexical Analyzer

- LRKPRS - LR(K) Parser
- R0CELL - Split Out Fields in Read State
- NXTSYM - Get Next Symbol
- LEXAN - Lexical Analyzer
- CHRGEN - Character Generator
- ADDCHR - Add Character to a String
- LOOKUP - Lookup String in a Table
- EXTSCN - Process Real and Integer Numbers
- ERROR - Parser Error Recovery
- APCEL 1 - Split out fields of apply state 1
- APCEL 2 - Split Out Fields of Apply State 2
- LACELL - Split Out Fields of Look-Ahead State
- CARD - Read Card/Print Card

e. Red Threat Planning

PLAN - Controls Red Threat Planning
THTRPLN - Red Theater Planner
CORBOUN - Creates Corridor Boundary Structures
REVISE - Revises Red Force Allocations
ABVSCOR - Matches Airbases with Corridors
CLOSCOR - Finds Closest Corridor to an Airbase
CANDTGT - Considers Candidate Targets
JGESUIT - Determines Geographic Suitability of Targets
FORMTGT - Assigns Formations to Targets
TGTGONE - Changes Target Status to Now-Available
AVAILBL - Changes Target Status to Available
SCHEDUL - Controls Flight Scheduling
RENDEVU - Calculates Formation Rendezvous Points
ACFRAG - Sets-Up Flight Frag's
CRFLTML - Creates Flight Data Structures
FLTGEOM - Determines Flight Mission Geometry
PLANOUT - Outputs Results of Planning Process
RLRAID - Releases Raidblok Structures
RLWAVE - Releases Wave Structures
RLTGTYP - Releases Target Type Structures
RLFMAKT - Releases formation Structures
RLCORD - Releases Corridor Structures
RLABDB - Releases Airbase Structures

f. Common Utilities

ADDBLOK - Adds a Data Block to the Top of a Linked List
CREATE - Creates Unit SB and C2 Data Structures
DROPBLOK - Drops a Block from a Singly Linked List
FINDBLOK - Finds a Block on a Singly Linked List
GETHEX - Finds or Creates a Hex Block
GIMME - Allocates Storage Space
HEXADD - Adds To Hex Numbers
HEXCHZ - Chooses Next Hex for a Flight to move to

HEXDIST - Calculates Distance Between Two Hexes
 HEXINV - Calculates Inverse of a Hex Number
 HEXMULT - Hex Multiplication by a Single Digit
 HISTORY - Records an Event for Prost Processing
 HXDGTS - Places a Hex Number in the IDIGITS Array
 LINEX - Determines Intersection of Two Lines
 LNPLOT - Prints a Line on the Output File
 MESSAGE - Prints a Message on the Output File
 OPTPTH - Finds Shortest Path Between Two Hexes
 PAGE - Advances Page of Output
 PELADD - Adds Target Blocks to the Potential Target
 Tree
 PTREE - Extracts a Block From the Target Tree
 PTRMRG - Sorts Target Tree
 RELEASE - Releases Storage Space
 RELIST - Subdivides Large Released Blocks
 RITEI - Prints an Integer Value
 RITEP - Prints a Pointer Value (octal)
 RITER - Prints a Real Value
 THH2PS - Translates Hex to Point Slope
 THX2XY - Translates Hex to X,Y Coordinates
 TLL2HX - Translates Lat-Long To Hex
 TXY2HXL - Translates X,Y to Hex at Specified Level
 PACK - Packs Two Halfword Fields (non-midas)
 VNPACK - Unpacks Two Halfword Fields (non-midas)
 NOWUCIT - Hex Peeper List Update
 GETPIRS - Pointer Retrieval
 KOMPARE - Compares Two Packed Words and Stores a New
 Value in the Specified Field.

g. Block Data Routines

BDALT - Semantic Processing
 BDLEX - Semantic Processing
 BDLRK - Semantic Processing

BDPARS - Semantic Processing
BLKDAT - Initialization

3. Main Processor Functions And Subroutines

The following lists summarize the functions carried out by various main processor subroutines. For a complete calling hierarchy and detailed subroutine documentation see Appendices I and J and the Source Code. Although They are not repeated in this section, all of the Red Threat planning functions and Subroutines found in the preprocessor are also resident in the main processor.

a. Top Level Control Routines

MADEM - Main Routine
FETCH - Reads Hold Files
CONTROL - Event Control
SELECT - Selects Appropriate Event Module
ASSIGN - Controls CRC Assisgment of Interceptions
ATTACK - Control Red Flight Ground Attacks
COMMO - Controls Communications
DOGFITE - Controls Air to Air Combat
ENGAGE - Controls Surface to Air Engagement
FLY - Controls Aircraft Movement
NAYBOR - Controls Determination of Unit Proximity
PERCEPT - Controls Perception of other Units
PLAN - Controls Red Threat Planning
PONDER - Controls Unit Information Processing
TOWER - Controls Airbase Operations
UMPIRE - Controls Simultaneous Event Processing

b. CRC Assignment Of Interceptions To Red Flights

ASSIGN - Controls CRC Assignments of Red Flights
INTASIN - CRC Attempts Interceptor Assignment
TGTHEX - Calculates Interception Point for Two Flights
BTNASIN - CRC Attempts BOC Assignment

c. Red Flight Group Attack Process

ATTACK - Controls Red Flight Ground Attacks
SHRKILL - Shorad Engaged Flights
DESTROY - Actions Carried Out When Unit is Destroyed
TERMACQ - Terminates Acquisition Devices and Peeper
list
KILFLIT - Actions Carried Out Flight Dies
FLTWYPE - Wipes Out Flight Action Order List
SAMWYPE - Actions Carried Out When A Sam Unit Dies
BYENOPS - Battery End Of Flight-Pass Up Procedures
SEEKP - Seek a Track in Perceptions List
CRCLOSS - CRC Actions On Loss Of Target Condition
BNLALLE - BOC Ponders Loss of All Engagement Possibil-
ities
HANDZPT - Handles Zero Priority Tracks in Adil
BATCEAS - Battery Decisions To Cease Engagement
CANCALO - Cancel Allocations
READIL - Return Track to Adil
RELOCAT - Reposition Entry in Doubly Linked List'
SEEKTFV - Seek New Engagement For a Fire Unit
TRKCHK - Air Defense Unit Ponders Trackability of TGT.
TOADIL - Refills Adil
ALLOFU - Allocate Fire Unit Against Red Flight
ALLOPAT - Allocate Patriot Against Red Flight
DILOUT - Eliminate Oil Entry
CNACTIK - Cancel Pending Actions for Track
CRCDIES - Actions Carried Out When CRC Dies
NUKBLND - Commo Block Out Of Units Near Nuke Blast

d. Communications

COMMO - Controls Communications Processes

e. Air To Air Combat

DOGFITE - Controls Air-To-Air Combat Processes
DESTROY - Actions Taken When Unit is Destroyed

f. Surface To Air Engagement

ENGAGE - Controls Surface-to-Air Engagement
TRYSHOT - Determine if a Missile can be Fired
BATCEAS - Battery Decision to Cease Engagement
BNLALLE - BOC Ponders Loss of all Engagement Possibilities
FIRECHK - Fire Missile Operations
AMMOCHK - Account for Missiles
BYNOTRO - Battery not ready for action
DILOUT - Eliminate Oil Entry

g. Aircraft Movement

FLY - Controls Aircraft Movement
VOLLOAD - Places a Unit on a Hex's Unit Occupancy list
FVELCHK - Checks Fuel Level
FLTWYPE - Wipeout Flight Action Order List
SKRKILL - Shurad Engages Flights
INTRFLY - Process Flight Actions at Decision Points
INTRFLY - Interceptor Flight Decisions
TGTHEX - Calculate Interception Point for Two Aircraft
COMMAND - Flights carry out orders
DESTROY - Action taken when Unit is Destroying

h. Determination Of Unit Proximity

NAYBOR - Controls Determination of Unit Proximity

i. Unit Perception

PERCEPT - Controls Unit Perception Process
ABSEE - Airbase Perception
CFLYCRC - CRC Perceptions Module Control
CRCSEE - CRC Message and Event Perception
CRCEVNT - CRC Direct Perceptions
DETECT - Detection Logic between Two Units
LOSRADR - Radar Line of Sight Determination
CRCTRAK - CRC Actions on Enemy Detection
CRCKIL - CRC Records Death Report

- BABMOVE - CRC Records Red Flight Movement
 - TGTHEX - Calculate Interception Point for Two Flights
 - NEWMOVE - CRC New Flight Perception Actions
 - FLYSEE - Flight Perception Module Control
 - ATKASES - Air Attack Damage Assessment on Ground Targets
 - CRCZINT - Interceptors Receive Messages from CRC
 - RONDSEE - Flight Perception of Rendezvous
 - GNDLOOK - Flight Perception and Attack of Ground Targets
 - SAMSEE - Sam Perceptions Module Control
 - SAMPRCM - Sam Perceives Aircraft Movement
 - BYTKCHK - Battery Tracking Operations
 - TOADIL - Try to Refill Adil
 - NEWPERC - Create Perceptions List Entry
 - BYPASUP - Battery Passes Track up to BOC
 - BYENDPS - Battery End of Flight-Pass Up Procedures
 - CRCLOSS - CRC Actions on Loss of Target Condition
 - DROPPQS - Battery Considers Drop Possibility
 - SEEKENG - Seek New Engagement
 - DROPPS2 - Battery Considers Drop Possibility 2
 - DLYACT - Add Action to Delayed Action Queue
 - ALLOBAT - Allocate BYRY against Red Flight
 - PRIORITY - Calculate Priority of Track
 - BNLALLE - BOC Ponders loss of All Engagement Possibilities
- j. Red Threat Planning
See Section (IIF.2)
- k. Unit Information Processing
- PONDER - Controls Unit Information Processing
 - TFLYCRC - CRC/Flight Ponder Module Control
 - CRCTHNK - CRC Decisions on Receipt of Information

CRCKIL	-	CRC Records Death Report
CRCTRAK	-	CRC Actions on Enemy Detection
CRCLOSS	-	CRC Actions on Loss of Target Condition
AB2CRC	-	CRC Interprets Message from Airbase
INT2CRC	-	CRC Interprets Message from Flight
DETECT	-	Detection Logic Between Two Units
BTN2CRL	-	CRC Interprets Message from BOC
AIRTHNK	-	Ponder Decision to go into Air Combat
DOGTHNK	-	Ponder Next Actions After Air to Air Combat
GOTOAB	-	Actions on Return Flights to Airbase
FUELCHK	-	Check Flight Fuel Level
BOCTINK	-	BOC Ponders Situation
BNPONSS	-	BOC Ponders Subordinate Status Message
BNPONBB	-	BOC Ponders BTRY Back In Action
FILERUP	-	Refuel Aircraft
INRANGE	-	Determine Time Period that Target is in Range
AZILIM	-	Inpose Azimuth Limits on Engage Window
INSECT	-	Calculate Intersection with Sector Limits
SETASSN	-	Set Up for Possible ADV. Assignment
PRIORITY	-	Calculate Priority of Track
SEEKENG	-	Seek New Engagement
DLYACT	-	Add Action to Delayed Action Queue
BYUPDAT	-	Update Battery Status
BNNOTRD	-	BOC Not Ready For Action
DILOUT	-	Eliminate DIL Entry
BATTOUT	-	BTRY Decisions to Drop an Enemy Track
BNLALLE	-	BOC Ponders Loss of All Engagement Possibilities
CHKLAST	-	Determine Last Chance for Engagement and Set Priority
DROPPOS	-	BTRY Considers Drop Possibility
COVAPLY	-	Cover Threat Allocation Decisions

SEEKTAC	-	Seek to Assign Coverage
ALLOBAT	-	Allocate BTRY against Red Flight
READIL	-	Return Track to Adil
DECRALO	-	Decrease Allocations
BNPONEP	-	BOC Ponders Engagement Progress
SKSBTRK	-	Seek Reporting Subordinate and Track
BNCMDPR	-	BOC Command Decision Processes
ACCEPT	-	BTRY Accepts Assignments
BYALCOV	-	BTRYs Alter Coverage Level
CANCALO	-	Cancel Allocations
SEEKTFV	-	Seek New Engagement for a Fire Unit
PATDEC	-	Patriot Decrease Coverage on Track
BATTCOV	-	BTRY Decision to Allocate Coverage on Track
TRKCHK	-	Ponder Trackability of Flight Target
ALLOFV	-	Allocate Fire Unit Against Red Flight
DLYACT	-	Add Action to Delayed Action Queue
BYHEDUP	-	BTRY Handle Engagement Data Update
SEEKP	-	Seek Track in Perceptions List
BNPONFD	-	BOC Ponders Flight Attrition
SDIGEST	-	Sam Process Information
BNNWTRK	-	BOC Handles Newly Visible Track
CHKCOV	-	Check for Coverage
BYNWTRK	-	BTRY Handles Newly Visible Track
PREPAFU	-	Prepare for Allocation of Fire Units
AUTOPRI	-	BTRY Decision of Target Priority
BNCONTC	-	BOC Considers Change in Track
BNCONHD	-	BOC Considers New Heading
BNRECOV	-	BOC Considers Reducing Coverage
BYCONTC	-	BTRY Considers Change in Track
BYCONLS	-	BTRY Considers Loss of Track
BATCEAS	-	BTRY Decision to Cease Engagement
BNPONDA	-	BOC Processes Delayed Actions

BNPNBD - BOC Ponders BTRY Death
 SAMATON - BTRY Autonomous Actions
 BTRYTNK - BTRY Ponders Situation
 BYCMDPR - BTRY Command Decision Processes
 BYPONTM - BTRY Ponders Track Movement
 BYPONER - BTRY Ponders Engagement Results
 PTPONER - Patriot Ponders Engagement Results
 BYPONFO - BTRY Ponders Red Movement
 BYPONRL - BTRY Ponders Reload
 BYPONRS - BTRY Ponders Resupply
 RESUPPLY - BTRY Ponders Resupply of Fire Units

1. Airbase Operations

TOWER - Controls Airbase Operations
 VOLLOAD - Places Unit on Hex Unit Occupancy List
 INLTACQ - Initialize Acquisition Devices
 REDEBRF - Process Flight After Landing
 WIPEOUT - Invalidated a Node on Leftist Tree
 DESTROY - Action When Unit is Destroyed
 GOGETEM - Launch Interceptor Flights
 CRFLTML - Create Flight Structures
 PTRAND - Generate Uniform Random Hex
 FLTGEOM - Calculate Flight Geometry

m. Simultaneous Event Control

UMPIRE - Controls Simultaneous Events
 DESTROY - Action When Unit is Destroyed
 NUKBLND - Commo Black out of Units Near NUKE Blast

Common Utilities

ADDBLOK - Adds a Data Block to the Top of Linked List
 CREATE - Create Unit SB and C2 Data Structures
 DELADD - Discrete Event Tree Adder
 DGTSHX - Converts a Value to a Hex Number
 DROPBLK - Drop a Block from a Singly Linked List

FETCH	-	Red in Hold Files
FINDBLK	-	Find a Block in a Singly Linked List
FSDUMP	-	Print Out ISPACE Contents
FSINIT	-	Free Space Control Initialization
GETHEX	-	Find or Create a Hex Data Block
GIMME	-	Allocate Storage Space
HEXADD	-	Adds Two Hex Numbers
HEXCHZ	-	Chooses Next Hex for a Flight to move to
HEXDIST	-	Calculates Distance Between Two Hexes
HEXINV	-	Calculates inverse of a Hex Number
HEXMLT	-	Hex Multiplication
HEXMOVE	-	Hex Movement
HEXMULT	-	Hex Multiplication
HISTORY	-	Records an Event for the Postprocessor
HXDGTS	-	Places a Hex Number in the IDIGTS Array
HXMLT2	-	Hex Multiplication
IJ2HX	-	Converts I,J Coordinates to Hex
JUGGLE	-	Ships Hex Number to Usable Order
UNEX	-	Determines Intersection of Two Lines
LNPLT	-	Prints a Line on the Output File
LOADPL	-	Loads the Players List
LTREE	-	Merges a Node on the Leftist Tree
LTRMRG	-	Merges a Node on the Leftist Tree
MESSAGE	-	Prints a Message on the Output File
PACK	-	Packs Two Halfword Fields (non-midas)
PAGE	-	Adds Target Blocks to the Potential Target Tree
PELADD	-	Adds Target Blocks to the Potential Target Tree
PTREE	-	Extracts a Block from the Target Tree
PTRMRG	-	Sorts Target Tree
RITEI	-	Prints an Integer Value

RITEP	-	Prints a Pointer Value
RITER	-	Prints a Real Value
UNPACK	-	Unpacks Two Halfwords (non-midas)
UNLINK	-	Removes a Dead Unit from the C2 Tree
YANK	-	Removes an Entry from a Doubly Linked List
STICK	-	Inserts an Entry in a Doubly Linked List
NOWUCIT	-	Hex Peeper List Update
GETPTRS	-	Gets Required Pointers
MESBILD	-	Builds a Message Structure
UNSTAT	-	Unpacks an Aircraft Flight Status Board into a Common
STATPAK	-	Repacks an Aircraft Flight Status Board after it has been Updated
SSL	-	Sorts a Singly Linked List
INSERT	-	Inserts an Entry into an Ordered Linked List
RELOCAT	-	Reposition Entry in Doubly Linked List

4. Postprocessor Functions And Subroutines

The following list summarizez functions carried out by various postprocessor subroutines. For a complete calling hierarchy and detailed subroutine documentation see Appendices I and J and the Source Code.

(1)	RECORD	-	Main Control Routine
(2)	INDEX	-	Finds Desired Index in a Storage Array
(3)	MESSAGE	-	Prints a Message on the Output File
(4)	PAGE	-	Advances Output Page
(5)	RITEI	-	Prints an Integer Value
(6)	RITER	-	Prints a Real Value
(7)	TABOUT	-	Prints Contents of Storage Arrays

CHAPTER III
DATA STRUCTURE DOCUMENTATION

A. INTRODUCTION

MADEN uses a list processing system to store data. Information is stored in the dynamically allocated array ISPACE in a series of data structures. These data structures are composed of blocks linked in a variety of configurations (eg., trees, lists, etc.) by pointers. Each block is made up of a specified number of words. In many cases words are divided into fields which contain discrete pieces of information. The pointers which connect blocks into data structures are actually addresses in the one dimensional array ISPACE. Since the number and configuration of blocks stored in ISPACE changes during model execution, the storage system is said to be dynamic. ISPACE is currently limited to 131,000 words of storage.

Sections B through G of this manual document the common blocks and data structures used by MADEM. Table III-1 provides a cross reference of data block types and the sections of this manual in which they are discussed. Since these blocks are defined and accessed using the Modular Information Data Access System (MIDAS), documentation of MIDAS and the tables used to define the data blocks have been included in Appendices B and D. In addition, Table III-2 provides an outline of the standard data structure documentation format used in this manual.

TABLE III-1. DATA BLOCK DOCUMENTATION CROSS REFERENCE

Block Name	C DATA BASE	D1 EVENT LISTS	D2 HEX TREE	E1 PLAYER LISTS	E2 C2 TREES	E3 PASSIVE TGT	F1 RED CMDR	F2 POTENTIAL TGT	F3 ASSIGNED TGT	F4 NONAVAIL TGT	F5 CORRIL-OR	F6 ATK REQ	F7 RED AB	F8 RED FLT	G1 ATAF	G2 SOC	G3 CHC	G4 BLUE AB	G5 BLUE FLT	G6 BOC	G7 BTRY	G8 PASSIVE TGT	H STORAGE
ABINFO																							
ABQEDB																							
ABSTATUS																							
ABVCR																							
ACDB																							
ACORUF																							
ACQ DEVICE																							
ACRF LIST																							
ACRFONAB																							
ACTAB																							
ADLINK																							
ADSTEDB																							
ALLOCATE																							
ADDB																							
ARCFSAVV																							
ARCFSTATUS																							
ATTACK BLOK																							
BOCSTAT																							
BRYDIL																							
BRYSTAT																							
BTYPSTAT																							
C2 BUFFER																							
C2 SIDE																							
COMMAND																							
CONST BLOK																							
CORRIDOR BLOK																							
CRCSEEBLOK																							
CRCSEEBLUE																							
CRCSEERED																							
CRCSEES																							
CRCSEEBLOK																							
DAQE																							
CRCSUBORD																							
DAT BLOK																							
DAT BUF																							
DCLASS BLOK																							
DIB																							

TABLE III-1. DATA BLOCK DOCUMENTATION CROSS REFERENCE (CONTINUED)

	DIL	DUMMY BLOCK	ENGAGE	EYE BALL	FAKTG BLOCK	F0BD BLOCK	FIRE UNIT	FLTAKI BLOCK	FLTDB	FMFLTDB	FOREST	FORMATION	FORMATION BLOCK	FORMATION BUFFER	HEAD BLOCK	HEXBLOCK	HEXELV	HEXLINK	LEFTREE	LINK	LOAD	MESSAGE	MUN	NOAVAIL BLOCK	ORDERS	PAT	PAYBUF	PAYDOBLOK	PAYLOAD	PER LIST	PLAYER BUFFER	PL BUFFER	PL YLST		
C DATA BASE																																			
D1 EVENT LISTS																																			
D2 HEX TREE																																			
E1 PLAYER LISTS																																			
E2 G2 TREES																																			
E3 PASSIVE TGT																																			
F1 RED CMDR																																			
F2 POTENTIAL TGTS																																			
F3 ASSIGNED TGTS																																			
F4 NONAVAIL TGTS																																			
F5 CORRIDOR																																			
F6 ATK REQ																																			
F7 RED AB																																			
F8 RED FLT																																			
GTATAF																																			
G2 SOC																																			
G3 CRC																																			
G4 BLUE AB																																			
G5 BLUE FLT																																			
G6 BOC																																			
G7 BTRY																																			
G8 PASSIVE TGT																																			
H STORAGE																																			

TABLE III-1. DATA BLOCK DOCUMENTATION CROSS REFERENCE (CONTINUED)

	PROFLED BLOC	PR BUFFER	QUESTAT	QUESTAT	RAID BLOC	READY QUE	SB	SDB	SEE BUF	SEER	SOURCE	STD BLOC	SUB	SUBLIST	SUBTYPE	TARGET BLOC	TARGET BUFFER	TGT LIST BLOC	TTD BLOC	WAVE BLOC	WINGMAN	GARBAGE LIST BLOC	
C. DATA BASE	●																						
D.1 EVENT LISTS																							
D.2 HEX TREE																							
E.1 PLAYER LISTS																							
E.2 C2 TREES																							
E.3 PASSIVE TGT																							
F.1 RED CMDR																							
F.2 POTENTIAL TGTS																							
F.3 ASSIGNED TGTS																							
F.4 NONAVAIL TGTS																							
F.5 CORRIDOR																							
F.6 ATTK REQ																							
F.7 RED AB																							
F.8 RED FLT																							
G.1 ATAF																							
G.2 SOC																							
G.3 CRC																							
G.4 BLUE AB																							
G.5 BLUE FLT																							
G.6 BOC																							
G.7 BTRY																							
G.8 PASSIVE TGT																							
H. STORAGE																							

4J68/75W

TABLE III-2. STANDARD DATA STRUCTURE DOCUMENTATION FORMAT

*	<u>SECTION HEADING</u>
A.	<u>DATA BLOCK INDEX</u> A list of MIDAS data blocks discussed in the section.
B.	<u>DESCRIPTION</u> A general description of the data structure and its purpose in MADEM. This description refers to the diagrams in subsections C and D. Other diagrams which show what the structure is attempting to represent in the real world (ex: 4 C2 hierarchy) are also included if they are helpful to an understanding of the structure.
C.	<u>STRUCTURE OVERVIEW</u> This subsection is intended to give the user a general view of the structure and its' component blocks. 1) <u>STRUCTURE DIAGRAM</u> A diagram of the entire structure showing the overall configuration of data blocks. 2) <u>BLOCK DEFINITIONS</u> Short definitions of all data blocks in the structure.
D.	<u>BLOCK SPECIFICATIONS</u> This subsection is intended to give the user a detailed view of the data blocks in the structure and the exact definition of variables in each field. 1) <u>BLOCK DIAGRAMS</u> Detailed diagrams of all data blocks including field names. 2) <u>FIELD DEFINITIONS</u> Precise definitions for all fields in all blocks. These definitions are sub-divided by block and contain all relevant values.
E.	<u>LINKAGES TO OTHER DATA STRUCTURES</u> Linkages to other data structures are noted along with the purpose of the linkages.
F.	<u>NOTES</u> Any notes or questions relating to the data structure and its' contents.

B. COMMON BLOCKS

- (1) COMMON/AAPK/ - air-to-Air PK Common
 - AAPK - antiaircraft probability of kill
- (2) COMMON/ACFRAG/ - AIRCRAFT FLIGHT Plan Common
 - NSECTOR - Corridor Zone Number
 - PTGTHEX -
 - PADRCG -
 - PHEXENT -
 - TATREND -
 - PHEXEXT -
- (3) COMMON/AFM/ - Debugging common for ?TRACE? JINX -
- (4) COMMON/AGPD/ - Air to Ground Probability of Detection AGPD -
- (5) COMMON/AGPK/ - Air to Ground Probability of kill AGPK -
- (6) COMMON/CLOOK/ - Common for dumping COMMONS
 - ISTABL -
 - NRSRV -
 - ISTAB -
- (7) COMMON/COMOUT/ -
 - LBL -
 - LBLCT -
- (8) COMMON/COMPTR/ -
 - PHXTOP - pointer to top of address tree
 - PTRBLUE - pointer to top of blue C2 tree
 - PTRRED - pointer to top of red C2 tree
 - PBLTGT - pointer to top of list of red targets
 - PTRC2 - pointer to active C2 tree
 - PBLUPL - pointer to blue players list
 - PREDPL - pointer to red players list
 - PTRPL - pointer to active players list
 - PLASTC - C2 pointer of last mentioned commander
 - PLASTS - C2 pointer of last mentioned subordinate
 - LSIDE - Active side (1 = Blue, 2 = red)

LHEX	-	last hex mentioned
PTRCDB	-	last mentioned corridor description pointer
PTRWDB	-	last mentioned wave description pointer
PTRRDB	-	last mentioned raid description pointer
PTRTTDB	-	last mentioned target type description pointer
BUFZN	-	buffer zone width value
IGMSRT	-	game hour start time, default = 1
IDYSRT	-	game day start time, default = 1
PTRDATA	-	pointer to data base
PTOPORD	-	pointer to red orders
PREDSEE	-	pointer to red commanders perception list
ISIZE	-	size of player list array, equal to maximum number of players on largest side
(9) COMMON/COMSCS/ -		
LMASK	-	bit mask for left hand side of a ward
RMASK	-	bit mask for right hand side of a ward
INITCEL	-	initialize the cell (currently not used)
TREETOP	-	top of the discrete event list
ENDTIME	-	end of the game
INCTIME	-	increment of time for interim output
GTIME	-	game of current discrete event
CELMINT	-	increment time for cell event
LPTR	-	array of pointers to cell events
(10) COMMON/CRCCOM/ -		
IEVENT	-	CRC's current event
PSB	-	pointer to the scoreboard
PADR	-	pointer to hex address
ITYPE	-	unit type
ISIDE	-	multiple definitions
(11) COMMON/CSTK/ -		
IPSTK	-	stack used by language processor
L	-	
LLMAX	-	
NSTK	-	

- (12) COMMON/DATA/ -
PTRDAT -
- (13) COMMON/DEBUG/
DBGA -
DBGB -
DBG -
DBGD -
- (14) COMMON/DFLAGS/ -
IDEBUG - a flag which turns on ENTRYR and EXITR. If
equal to 2HON turns on entry and exit messages
for all routines
IDUMP - ISPACE dumping option flag, set as a parameter
in MADEM if set equal to 2HON, ISPACE will be
dumped
IDATFLG - DATFILE display flag, set as input parameter
in MADEM 2HON - will display DATFILE data
data structure only in INITBIN
ISTOP - input parameter set in MADEM
- if equal to 4HODAT stops after UOIL (can be used
to get ISPACE dump before players list is
released)
- if equal 3HDEL stops INITBIN after DELADD and
before CONTROL
- (15) COMMON/XTRACE -
ITRPTR - Pointer to circular list
TRACIR - circular list of last 50 routine entries
TRAPTR - pointer to routine pushdown stack
TRAPDS - push down stack of routine entries
ICOUNT - vector of routine entry counts
- (16) COMMON/STEXT/ -
SEGTX - vector of routine names in hollerith

(17)	COMMON/FS/	-	
	PTRFS	-	pointer to first ward of free space
	PTRRSL	-	pointer to recovered storage list
	MXSPACE	-	maximum length of ISPACE, set in MADEM
	ID	-	last Assigned Player ID Number
(18)	COMMON/HALT/	-	
	IFSTOP	-	a flag, if equal to 1, MADEM stops in control
	CPULIM	-	maximum number of CPU second set in MADEM
	CELTIME	-	
	LEVEL	-	
(19)	COMMON/INITPTR/-	-	All major values from Player SB.SDB
	PTRADR	-	PTR to current address
	PPTRC2	-	PTR to C2 node
	PTRSDB	-	pointer to status display board
	PTRFEL	-	pointer to future event list
	PTRACQ	-	pointer to ACQDEVICE block
	PTRCOM	-	pointer to command block
	PTRSTAT	-	Pointer to a unit status block
	PTRSEE	-	pointer to perceptions list
	PTRSUB	-	pointer to a subordinate block
	PTRORD	-	pointer to orders block
	MYTYPE	-	Player unit type
	MYSIDE	-	player side
(20)	COMMON/IODEV/	-	
	IN	-	device number of input file
	NUOIL	-	device number of UOIL file
	ND	-	device number of DATFILE
(21)	COMMON/JJOPT/	-	
	IOP	-	start or restart option flag
		-	if equal to 1 start at beginning (INITBIN)
		-	if equal to 2 restart (RUNBIN)

(22)	COMMON/LIMITS/	-	Max & Min limits on ISPACE
	LOWER	-	Extend core location of ISPACE (1)
	LUPPER	-	extended core location of last work of ISPACE
(23)	COMMON/MASK/	-	
	IL	-	masking constant = 77777777770000000000B
	ILM	-	masking constant = 100000000000B
(24)	COMMON/MODVAR/	-	
	INCNT	-	incident code (event code)
	NEHMEN	-	receiver of the event
	LASSEN	-	schedules of an event
	TIME	-	game time of the scheduled event
	MSG	-	Pointer to the message block
	PTRGOD	-	PTR to a god like creature whose omnipotence awes all who meet
	RSEED	-	random seed
(25)	COMMON/MXMIS/	-	
	MXSUP	-	
(26)	COMMON/PATH/	-	
	LASTP	-	
	LREP	-	
	LASTR	-	
(27)	COMMON/SAMPK/	-	
	SAMPKA	-	SAM probability of kill
	SAMPKB	-	SAM probability of kill
(28)	COMMON/SAMPTRS/-	-	
	PMYDATA	-	pointer to data base associated with considered unit
	PDIL	-	pointer to element in digested information list which corresponds to a particular considered flight

PPPINFOR - pointer to considered perception list entry
 for battery or BOC
 PDINFO - pointer to considered digested information
 block for battery or BOC
 POINFO - pointer to old digested information block for
 battery or BOC
 PBAT - pointer to SUBLIST block representing a par-
 ticular subordinate BOC
 PPAL - pointer to PAL entry for track
 PFU - pointer to fire unit block
 (29) COMMON/SEMINFO/-
 IVALUE - an array which holds real number from the
 SEMANT sentences
 IFLAG -
 (30) COMMON/SPACE/-
 BLANK - required!
 ISPACE -
 (31) COMMON/SPSTAT/-
 ICIGIM - counts the number of blocks by block size that
 GIMME allocates
 ICTREL - counts the number of blocks by block size
 that are released by RELEASE
 (32) COMMON/STATBD/- AIRCRAFT STATUS Board
 PFLTYP - pointer to Flight DATA BASE
 PTRMUN - pointer to MUNITIONS list
 PTRSTRT - pointer to start hex of leg
 PTREND - pointer to end hex of leg
 PTRNXT - pointer to next hex in flight path
 PTRAB - pointer to SB of home 3
 NUMTGT - number of aircraft in air-to-air track

	PTGTSB	-	pointer to air-to-air target SB
	PGNDTGT	-	pointer to air-to-ground target SB
	NUMAC	-	number of aircraft in flight
	LEGSTA	-	type of leg currently being blown
	INSTA	-	status of interceptor
	ITALTCNG	-	climbing, diving level
	NDXPROF	-	index into flight profile list
	LNDSTA	-	landing status
	IORBSTA	-	orbit status
	IAIRCOM	-	air-to-air combat status
	IGNDATK	-	air-to-ground combat status
	JAMSTA	-	jammer status
	FUEL	-	current fuel in of hexes
	ALTUDE	-	current altitude
	SPEED	-	current speed
	DIRECT	-	current heading
(33)	COMMON/THTRPLN/-		
	PFOREST	-	pointer to target list
	PNOAVAL	-	pointer to non available target list
(34)	COMMON/TRACK/	-	
	PFADR	-	
	ISID	-	
(35)	COMMON/TYPES/	-	
	IARRAY	-	
(36)	COMMON/OUTDEV/	-	
	OUTDEV	-	output device for data structure displays

C. DATA BASE STRUCTURE - DATFILE

1. Data Block Index

ACDB
ABQUEDB
ACRFTLIST
ACRFTONAB
ADSITEDB
AQDB
DATBLOK
DATBUF
FDBDBLOK
FLTDB
FMFLTDB
PAYBUF
PAYLOAD
PAYLDBLOK
PROFILEDBLOK

2. Description

The DATFILE data structure, built by subroutine OTHRDAT, contains only data from the MADEM input file 'DATFILE'. The DATFILE data is divided into 13 classes. Nine of these classes (classes 6026, 6001-6008) are stored in the DATFILE data structure. The other four classes are in common blocks.

The DATFILE data structure is a matrix of linked lists that includes 15 distinct data blocks. Fourteen of those data blocks point to themselves and thus are singly linked lists. The only block that is not part of a linked list is the one word buffer (DATBUF) at the top of the DATFILE structure.

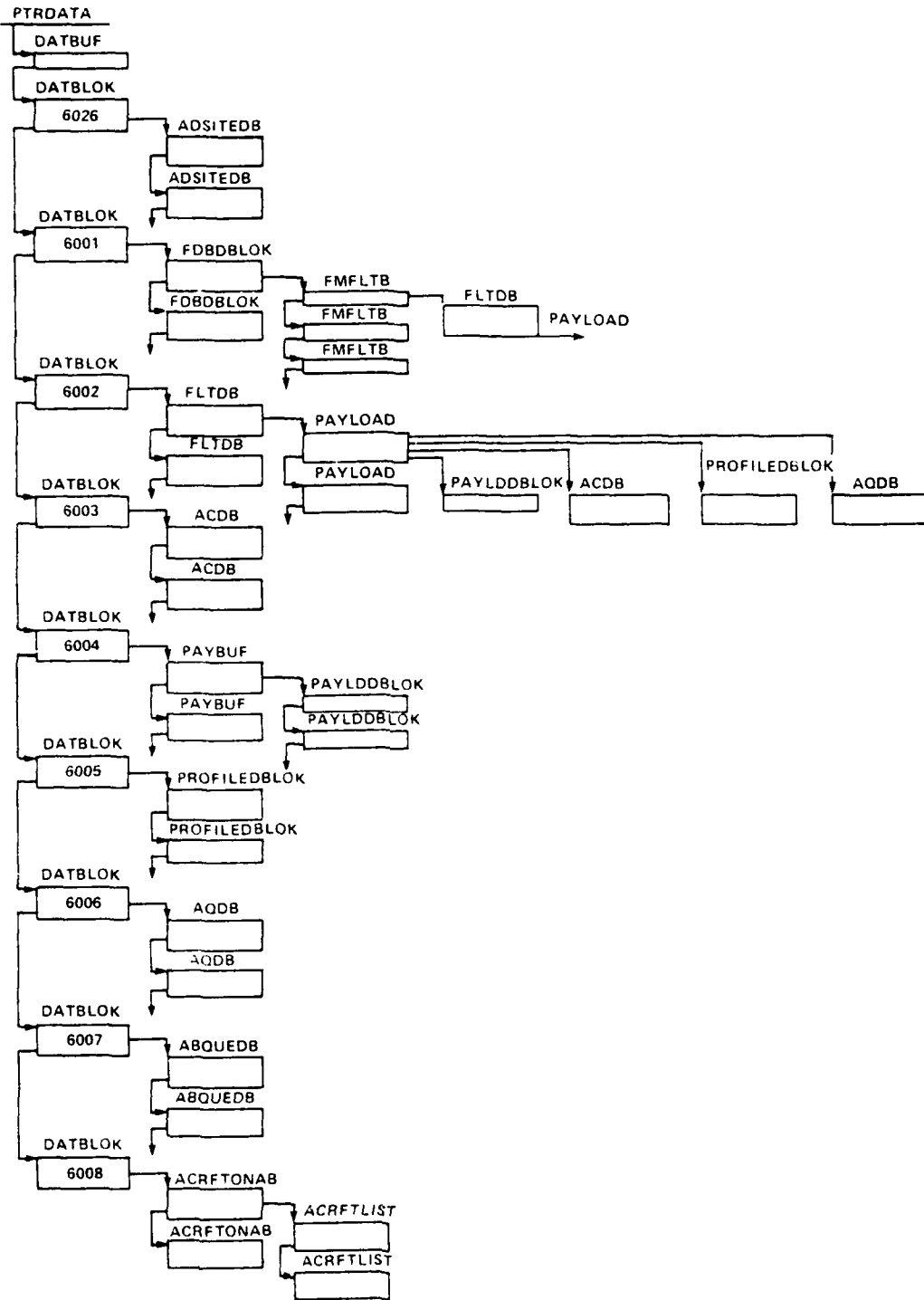
The main linked list in DATFILE consists of nine occurrences of DATBLOK. Each occurrence is a two-word buffer block for one of the nine classes of DATFILE data that are stored in DATFILE. Each two-word buffer points to a data block specific that class (6000 SERIES). Some of those data blocks point to even more linked lists. See the diagrams for a better picture of DATFILE.

3. Structure Overview

a. Structure Diagram (See Figure III-1)

b. Block Definitions

<u>DATBUF</u>	<u>Data Base Buffer Block.</u> Top of entire (DAFILE) structure. Pointed to by PTRDATA in common <u>COMPTR</u> .
<u>DATBLOK</u>	<u>Data Class Block.</u> Top of <u>DATBLOK</u> list pointed to by <u>DATBUE</u> block. One <u>DATBLOK</u> block for each data class (6000 SERIES).
<u>ADSITEDB</u> (Class-6026)	<u>Air Defense Site Data Base Block.</u> Used to describe BOC and Battery characteristics. One block for each type of BOC and Battery (e.g., HAWK, HERC, PATRIOT).
<u>EDDBLOK</u> (Class-6001)	<u>Formation Data Base Block.</u> Formation description. One block for each formation type.
<u>FMFLTDB</u> (Class-6001)	<u>Formations Flight Data Base Block.</u> Describes flights in the formation referred to in the <u>EDDBLOK</u> blocks.
<u>FLTDB</u> (Class-6002)	<u>Flight Data Base Block.</u> Contains basic flight characteristics. One block for each flight.
<u>PAYLOAD</u> (Class-6002)	<u>Payload Data Base Block.</u> Contains basic information on payload capacity and range.
<u>ACDR</u> (Class-6003)	<u>Aircraft Data Base Block.</u> Contains basic aircraft characteristics. One block for each aircraft type, Red and Blue.
<u>PAYBUF</u> (Class-6004)	<u>Payload Type Buffer Block.</u> Buffer block for <u>PAYLDDBLOK</u> blocks. One block for each payload type.
<u>PAYLDDBLOK</u> (Class-6004)	<u>Payload Identification Block.</u> Contains specific payload/weapon code. Represents a particular weapon of a given type.



4368,79W

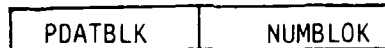
Figure III-1. Data Base Structure Diagram

<u>PROFILEDBLOK</u> (Class-6005)	<u>Profile Specification Block.</u> Describes basic mission profile in terms of altitude on various legs of the mission.
<u>AQDB</u> (Class-6006)	<u>Acquisition Data Base Block.</u> Specifies acquisition range for a given device.
<u>ABQUEDB</u> (Class-6007)	<u>Air Base Queue Data Base Block.</u> Service queue information for air bases. One for each air base.
<u>ACRFTONAB</u> (Class-6008)	<u>Initial Aircraft on Air Base Block.</u> Specifies number of aircraft types on an air base. One for each air base, Red and Blue.
<u>ACRFTLIST</u> (Class-6002)	<u>Aircraft List Block.</u> One for each aircraft type. Pointed to by <u>ACRFTONAB</u> . Contains aircraft type code.

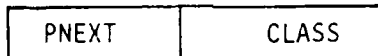
4. Block Specifications

a. Block Diagrams

1) DATBUF



2) DATBLOK



3) ADSITEDB

PNEXT		ADTYPE	
MODVAL1		MAXNUMDIGEST	
MAXTIMEDIGEST		MINTIMEDIGEST	
LOSTIME (SPACE)			
LASTCHANCE (SPACE)			
ENGAGEWINDOW(SPACE)			
MODVAL2 (SPACE)			
MODVAL3 (SPACE)			
COVONONE		ONE	
COVONFEW		FEW	
COVONMANY		MANY	
TIMEFLIGHT(SPACE)			
MISSILERANGE(SPACE)			
MAXASSIGN		MODVAL4	
MODVAL5(SPACE)			
MAXTRACTRANGE(SPACE)			
LOCKONTIME(SPACE)			
MODVAL6(SPACE)			
MODVAL7(SPACE)			
CONVLOAD			
SNUKELOAD		LNUDELOAD	
RESUPPLYCV		CVRESUPPLYFREQ	
RESUPPLYSN		SNRESUPPLYFREQ	
RESUPPLYSN		LNRESUPPLYFREQ	

4) FDBDBLOK

PNEXT		NRFORM	
PTRFLT		NOFLTL=3	
SPFORMC			

5) FMFLTDB

PNEXT		PNXFLDB	
-------	--	---------	--

6) FLTDB

PNXFLDB	NRFLITE
PTYPLDS	NOPYLDS
PTYAQDB	PTACDB
MAXNOAC	MINNOAC
MULTAC	PROFILE
SPFLTC(SPACE)	
DISTSEP(SPACE)	

7) PAYLOAD

PNXTYPD	NRPDCLS
MAXAMT	MINAMT
MAXFIRERANGE	PAYLDDB

8) ACDB

NEXT	NRACTYPE
MAXSPEED(SPACE)	
CRUISESPEED(SPACE)	
MAXALTITUDE(SPACE)	
MINALTITUDE(SPACE)	
MAXCLIMBDIVE(SPACE)	
FUELCONSUME(SPACE)	
ACQRANGE(SPACE)	
RADARCS(SPACE)	
ATTACKRADIUS(SPACE)	
MAXFUEL(SPACE)	

9) PAYBUF

PNEXT	NRPDCLS
PAYLDDB	NUMBLOK

10) PAYLDDBLOK

NEXT	TYPEINDEX
------	-----------

11) PROFILEDBLOK

PNXPRDB	NRPROFL
ALTCREW(SPACE)	
ALTTOTGT(SPACE)	
ALTDAB(SPACE)	

12) AQDB

NEXT	NRAQTYP
RANGE(SPACE)	
NOUSE1	
NOUSE2	

13) ABQUEDB

PNEXT	CLASS
VALUE1(SPACE)	
VALUE2(SPACE)	
VALUE3(SPACE)	

14) ACRFTONAB

PNEXT	ABID
ACRFTLIST	NUMBLOKS

15) ACRFTLIST

PNEXT	ACRFTID
NUMACRFT	
FORMTYPE	

b. Field Definitions

1) DATBUF Block

PDATBLK Pointer to first DATBLOK block.
NUMBLOK Number of DATBLOK blocks in list.

2) DATBLOK Block

PNEXT Pointer to next DATBLOK block.
CLASS DATFILE class (6000 SERIES).
 6026 = Air defense site data
 6001 = Formation specifications
 6002 = Flight specifications
 6003 = Aircraft specifications
 6004 = Payload specifications
 6005 = Flight profile specifications
 6006 = Acquisition device data
 6007 = Air base queue data
 6008 = Initial aircraft types on air
 bases

PCLASS Pointer to first data block for the
 class. The actual data block varies
 from class to class.

NUMBLOK Number of data blocks in list pointed
 to by PCLA.

3) ADSITEDB Block

PNEXT: Pointer to next ADSITEDB block.

ADTYPE: Unit type of this unit, must be a BOC or BTRY.

MODVAL1: Model Value = 1

MAXNUMDIGEST: Maximum number of flights on which
 a BOC or BTRY can be digesting info
 at one time.

MAXTIME DIGEST: Maximum time (in seconds) between
 consecutive digests of info (BOC &
 BTRY).

MINTIME DIGEST: Minimum time (in seconds) between
 consecutive digests of info (BOC &
 BTRY).

LOSTTIME: Time (in seconds) after which a track
 not seen is assumed permanently lost
 (BOC & BTRY).

LASTCHANCE: Time (in seconds) considered short
 for a subordinate to respond to a
 target. (Time from now until his
 last chance to shoot.) BOC only,
 for BTRY = 0.

ENGAGEWINDOW: Minimum length of subordinates
 engagement window for a significant
 engagement opportunity in seconds
 (BOC & BTRY).

MODVAL2: Model value = 0

MODVAL3: Model value = 0

COVONONE: Desired number of fire units coverage
 for one aircraft (BOC & BTRY).

ONE: Model value = 1
 COVONFEW: Desired number of fire units coverage
 for few aircraft (BOC & BTRY).
 FEW: Model value = 5, number of aircraft
 considered "few."
 COVONMANY: Desired number of fire units
 coverage for many aircraft (BOC &
 BTRY).
 MANY: Model value = 1000000.
 TIMEFLIGHT: Maximum time (in seconds of flight
 for missile (BOC & BTRY).
 MISSILERANGE: Maximum range for missiles in meters
 (BOC & BTRY).
 MAXASSIGN: Maximum number of targets per ready
 fire unit to be assigned at one time.
 BOC only, BTRY = 0.
 MODVAL4: Model value; for BOC = 8, for BTRY = 11.
 MODVAL5: Model value 0.
 MAXTRACKRANGE: Maximum tracking range in meters.
 BTRY only, BOC = 0.
 LOCK ON TIME: Assumed time (in seconds) for BTRY
 to achieve lockon. BTRY only, BOC
 = 0.
 MODVAL6: Model value = 0.
 MODVAL7: Model value = 0.
 CONVLOAD: Number of Conventional missiles.
 SNUKELoad: Number of large nukes.
 LNUKELoad: Number of large nukes.
 RESUPPLYCV: Number of missiles per resupply of
 ammo. BTRY only, BOC = 0.
 CVRESUPPLYFREQ: Time (in seconds) between resupply
 of conventional ammo. BTRY only,
 BOC = 0.

RESUPPLYSN:	Number of missiles per resupply of small nukes. BTRY only, BOC = 0.
RESUPPLYLN:	Number of missiles per resupply of large nukes. (BTRY only, BOC = 0.)
LNRESUPPLYFREQ:	Time (in seconds) between resupply of large nukes. BTRY only, BOC = 0.
4).	<u>ADSITEDB Block</u>
PNEXT:	Pointer to next FDBDBLOK.
NRFORM:	Formation number, must be unique.
PTRFLT:	Pointer to formations flight block (FMFLTDB).
NOFTL:	Number of flights in the formation.
SPFORMC:	Formation cruise speed in meters/seconds.
5)	<u>FMFLTOB Block</u>
PNEXT:	Pointer to next formation flight block.
PNXFLDB:	Pointer to flight data block (FLTDB).
6)	<u>FLTDB Block</u>
PNXFLDB:	Pointer to next FLTDB Block.
NRFLITE:	Unique flight specification number.
PTYPLDS:	Pointer to payload data block (Payload Class 6002).
NOPYLDS:	Number of payload data blocks.
PTYAQDB:	Pointer to acquisition data block (AQDB, Class 6007).
PTACDB:	Pointer to Aircraft Specification data block (ACDB, Class 6003).
MAXNOAC:	Maximum number of aircraft in flight.
MINNOAC:	Minimum number of aircraft in flight.
MULTAC:	Multiples of aircraft required for flight.
PROFILE:	Pointer to profile specification data block (PROFILEDBLOK, Class 6005).

SPFLTC:	Flight cruising speed in meters/seconds.
DISTSEP:	Flight separation distance in meters.
7) <u>PAYLOAD Block</u>	
PNXTYPD:	Pointer to next payload block.
NRPDCLS:	Payload type, must be 3 or 4: 3 = air to ground 4 = air to air
MAXAMT:	Maximum number of loads of this payload.
MINAMT:	Minimum number of loads of this payload.
MAXFIRERANGE:	Future use by an enhancement for maximum fire range for engagements greater than one hex.
PAYLDDB:	Pointer to payload ID DATA BLOCK (PAYLDDBLOK, Class 6004).
8) <u>ACDB Block</u>	
NEXT:	Pointer to next ACDB block.
NRACTYPE:	Aircraft type number.
MAXSPEED:	Maximum speed in meters/seconds.
CRUISESPEED:	Cruising speed in meters/seconds.
MAXALTITUDE:	Maximum altitude in meters.
MINALTITUDE:	Minimum altitude in meters.
MAXCLIMBDIVE:	Maximum climb/dive rate in meters/seconds.
FUELCONSUME:	Fuel consumption rate in hexes/seconds.
ACQRANGE:	Acquisition range in meters.
RDRARCS:	Radar cross section in hexes.
ATTACKRADIUS:	Attack radius in meters.
MAXFUEL:	Maximum fuel load in hexes.
9) <u>PAYBUF Block</u>	
PNEXT:	Pointer to next PAYBUF.

	NRPDCLS:	Type of payload, must be 3 or 4: 3 = Air to ground 4 = Air to air
	PAYLddb:	Pointer to ID Blocks (PAYLddbLOK, Class 6004) for this payload type. Number of ID Blocks for this payload type.
10)	<u>PAYLddbLOK Block</u>	
	NEXT:	Pointer to next ID block.
	TYPEINDEX:	Payload ID, unique within each payload type.
11)	<u>PROFILEDBLOK Block</u>	
	PNXPRDB:	Pointer to next PROFILEDBLOK.
	NRPROFL:	Profile Identification number, must be unique within the 6005 Class.
	ALTCREN:	Altitude of first leg in meters.
	ALTOTGT:	Altitude of second leg in meters.
	ALTOAB:	Altitude of third leg in meters.
12)	<u>AQDB Block</u>	
	NEXT:	Pointer to next AQDB block.
	NRAQTYP:	Unit type.
	RANGE:	Acquisition range in meters.
	NOUSE1:	Not used.
	NOUSE2:	Not used.
13)	<u>ABQUEDB Block</u>	
	PNEXT:	Pointer to next ABQUEDB block.
	CLASS:	Queue Class, currently model value = 1.
	VALUE1:	Model value = 0.005.
	VALUE2:	Model value = 0.
	VALUE3:	Model value = 0.2.
14)	<u>ACRFTONAB Block</u>	
	PNEXT:	Pointer to next ACRFTONAB block.
	ABID:	Air base ID.

ACRFTLIST: Pointer to aircraft list (ACRFLIST) for this air base.

NUMBLOCKS: Number of aircraft types on this air base. Limit 1 for Blue.

15) ACRFTLIST Block

PNEXT: Pointer to next ACRFTLIST block.

ACRFTID: Type of aircraft (400 Series).

NUMACRFT: Number of aircraft.

FORMTYPE: Formation type number.

* Required for Blue air bases.

* Equals zero for Red air bases.

5. Linkages to Other Data Structures

The DATFILE data structure is self-contained in the sense that none of its member data blocks point to blocks outside the DATFILE structure. The reason for this is that the DATFILE data structure is merely a representation of the input file DATFILE, and thus needs no external pointers.

Other data structures and blocks, however, point to DATFILE as follows:

<u>Block</u>	<u>Points To</u>	<u>Comments</u>
FORMATIONBLOK	FDBDBLOK	To identify each Red formation.
FLTAKTBLOK	FLTDB	Once for each Red flight attacking a target.
ACRFTSTATUS	FLTDB	Once for each Red aircraft to identify its flight.
LOAD	PAYLOAD	To identify an aircraft's payload.
ABINFO	ACDB	Once for each aircraft type. So that an air base has access to aircraft specs.
QUEUES	ABQUEDB	Air base queue data block.

The top of the DATFILE structure (DATBUF) is pointed to by PTRDATA, a variable in common block COMPTR.

6. Notes

Three of DATFILE's blocks--DATBUF, DATBLOK, and PAYBUF--were not in the pre-1979 "full" MIDAS table. As of August 20, 1979, however, plans were to include these blocks in the MIDAS table.

The third word of block PAYLOAD is defined differently in the MIDAS table than it was used in the non-MIDASed OTHRDAT routine. It was a full word pointer to PAYLDBLOK but has been redefined so that the 2nd half word is that pointer and the first half word is the maximum firing range for PAYLOAD. This was done to allow for a future enhancement.

D. SIMULATION CONTROL STRUCTURES

The simulation control structures are used to control event processing and movement within the hexagonal coordinate system. Event processing structures include the discrete event list and future event lists. Hexagonal coordinate system related structures include the hex address tree, unit occupance lists and peeper lists. These control structures are explained in detail in the following subsections.

1. Event Lists

a. Data Block Index

EVENT

LEFTREE

MESSAGE

b. Description

Event scheduling in MADEM is carried out using the three data block types listed above. These block types are used to build two event control data structures DEL and FEL.

The DISCRETE EVENT LIST (DEL) is the most complex of these structures. It uses all three block types to build a leftist tree of the form shown in the structure diagram under the DEL label. The DEL structure is used to track all scheduled events with the next event to occur placed at the top of the tree. The leftist tree form is used to speed sorting of the large number of events scheduled by the model. For a complete explanation of leftist tree sorting algorithms see Appendix E. It should be noted that the DEL is actually a threaded tree in that the event blocks have pointer fields which allow them to be independently linked into doubly linked lists. This capability is shown in the structure diagram by the dotted lines.

The future event list (FEL) is a doubly linked list formed within the DEL structure using the threading capability described above. The FEL is used to keep track of events scheduled by a particular player. This allows future events (as described by EVENT, LEFTREE, AND MESSAGE blocks) scheduled by a player to be deleted from the DEL with minimal effort if the player is destroyed before these events can occur. An example of a FEL is shown in the structure diagram.

c. Structure Overview

1) Structure Diagram (Figure III-2)

2) Block Definitions

LEFTREE - LEFTIST EVENT TREE NODE. Contains time of event used for sorting and pointers which connect the node to the rest of the tree.

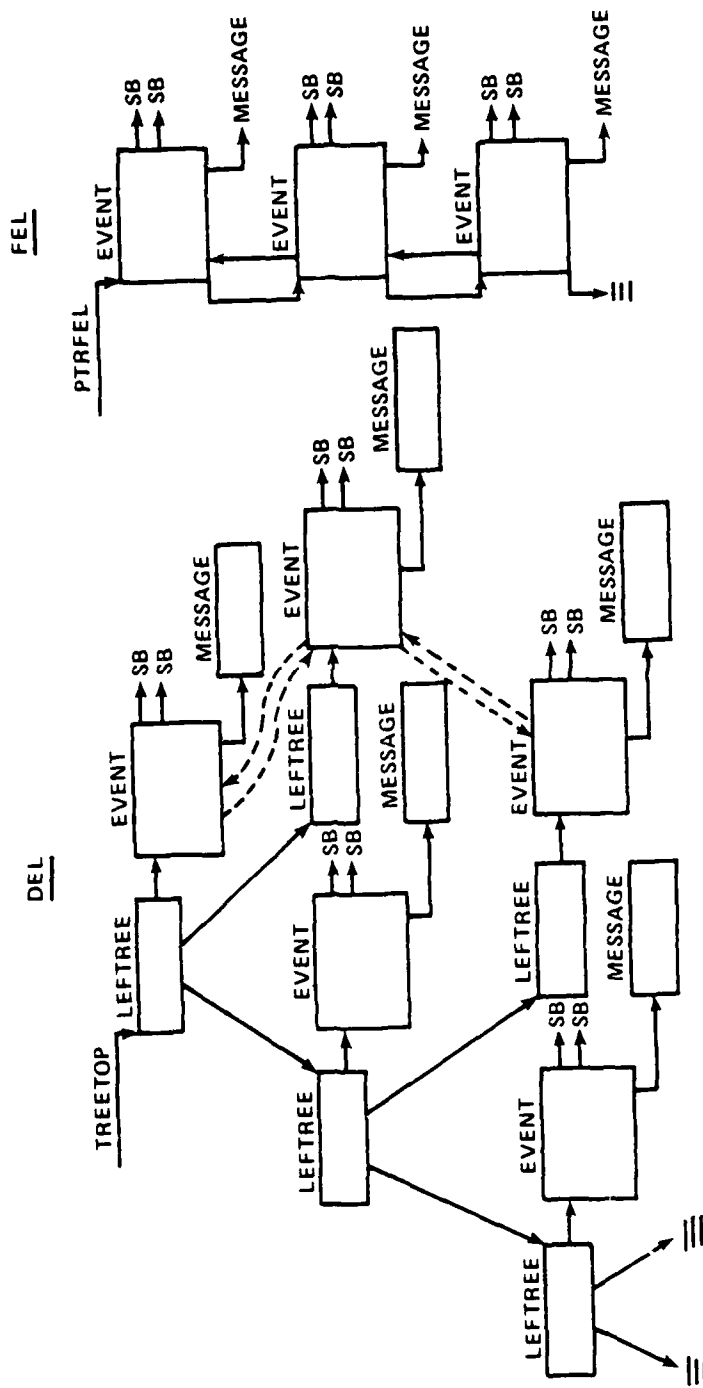


Figure III-2. Discreet Event List Diagram

4368/79W

EVENT EVENT BLOCK. Contains a description of the event, its time of occurrence, pointers to the SB (Scoreboards) of the perpetrator and victim of an event. MESSAGE. Contains message codes associated with an event.

d. Block Specifications

1. Block Diagrams

a. EVENT

NEHMEN	INCDNT
PTRUP	PTRDOWN
MSG	LASSEN TIME (SPACE)

b) LEFTREE

TIME (SPACE)			
PREVENT	DIST	PLEFT	PRITE

c) MESSAGE

PTR	FREQ
VALUE 1	VALUE 2
+ VALUE 3	

Note: + indicates alternate field definition for preceding word.

2. Field Definitions

a) EVENT BLOCK

- NEHMAN pointer to the player to be processed in the event. (Points to the player's scoreboard - SB)
- INCONT - Event code: (See Appendix F.)
- PTRUP - FEL pointer to preceding event block for the player scheduling the event.
- PTRDOWN - FEL pointer to following event block for the player scheduling the event.
- MSG - pointer to the MESSAGE block. Used only when a message event has been scheduled.
- TIME - Time of the event. Stored as a real variable.

b) LEFTREE Node

TIME - Time of the event. Stored as a real variable.
PEVENT - pointer to the EVENT block.
DIST - Minimum distance (in nodes) from a node (LEFTREE) to a leaf of the leftist tree. (Note: leaves are vacuous)
PLEFT - Pointer to LEFTREE node on the right of the DEL tree.

c) MESSAGE Block

PTR -
FREQ -
VALUE 1 -
VALUE 2 -
VALUE 3 -

e. Linkages to other Data Structures

The event blocks are linked to the unit scoreboards (SB) of both the player initiating the event and the player to be processed by the event. The DEL and FEL are also pointed to by the pointers TREETOR and PTRFEL, respectively. In addition, the EVENT blocks are pointed to be the following data blocks:

BOCSTAT	DIL	PAL
BTRYDIL	ENGAGE	SB
BTRYSTAT	FIREUNIT	

f. Notes

The MADEM event code definitions listed in Appendix F indicate the code number of each MADEM event and the subroutine in which event processing occurs.

2. Hex Address Tree and Related Lists

a. Data Block Index

BUFFER

HEX

HEXELEV

LINK

b. Description

Terrain information and the position of units in the hexagonal coordinate system is stored in the HEX ADDRESS TREE and its associated PEEPER and UNIT OCCUPANCE lists.

The HEX ADDRESS TREE is composed of HEX blocks arranged in a hierarchical tree structure. Each of the levels in the tree corresponds to a level in the hexagonal coordinate system. Table III-3 indicates the levels at which various types of information are stored in the HEX blocks terrain field as well as the levels at which PEEPER and UNIT OCCUPANCY lists are maintained.

UNIT OCCUPANCY and PEEPER lists may be attached to HEX blocks at various levels in the HEX ADDRESS TREE structure. These linked lists are composed of a BUFFER block and a chain of LINK blocks.

The UNIT OCCUPANCY LIST (UOL) contains pointers to the scoreboards (SB) of units which occupy the HEX at the specified level.

The PEEPER LIST (PL) contains pointers to the scoreboards (SB) of units which can SEE INTO the hex.

1) Structure Diagram (Figure III-3)

2) Block Definitions

HEX	-	HEX BLOCK. Contains HEX number, tree pointers, unit occupancy list and peeper list pointers and terrain pointer.
BUFFER	-	BUFFER BLOCK. Used by PEEPER and UNIT OCCUPANCY lists. Contains pointers to scoreboards (SB).

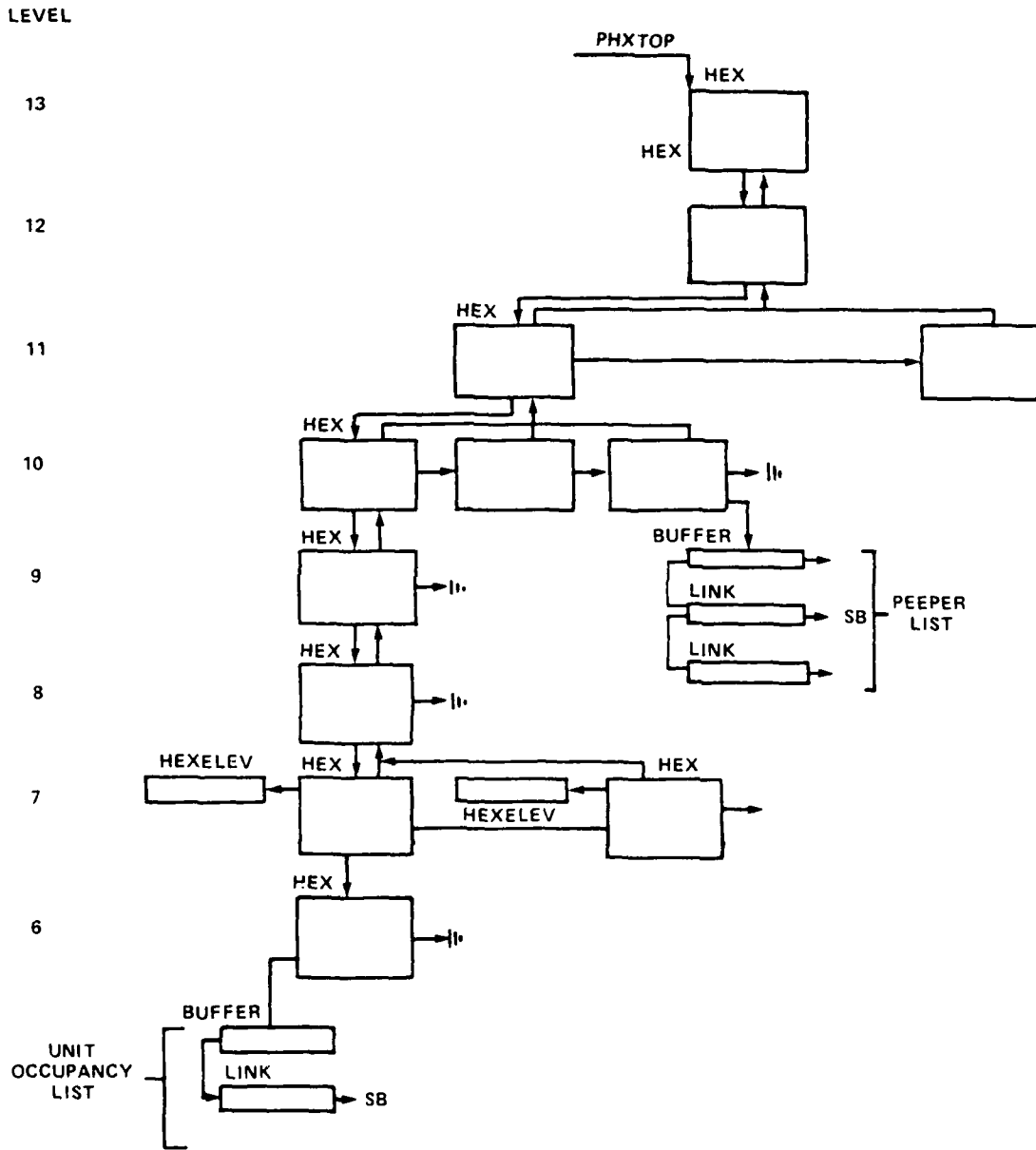


Figure III-3. Hex Tree Structure Diagram

4368/79W

TABLE III-3.
HEX BLOCK DATA SPECIFICATIONS

HEX LEVEL		NO. OF HEX DIGITS	HEX DIAMETER (KM)	HEX AREA $d^2 \quad 3/2$	STORAGE
DEC	OCT				
13	15	0	8575.	63,700,000	
12	14	1	3240.	9,100,000	
11	13	2	1225.	1,300,000	
10	12	3	463.	185,600	1
9	11	4	175.	26,500	
8	10	5	66.1	3,790	
7	7	6	25.	541	2
6	6	7	9.45	77	3

- 1 ● Pointer to the PEEPER list is active at this level.
- 2 ● Pointer to elevation storage block active at this level.
- Pointer to PEEPER list active
- Pointer to Unit occupancy list active
- 3 ● Pointer to Unit occupancy list active.

d. Block Specifications

1). Block Diagrams

a) HEX

HEXNUMBER	LEVEL
PUP	PDOWN
TERRAIN	PNEXT
PUOL	PEEPER

b) HEXELEV

ELEVAT (SPACE)

c) BUFFER

PSTART	NUMLINK
--------	---------

d) LINK

PNEXT	PSB
-------	-----

2) Field Definitions

a) HEX Block

- HEXNUMBER - HEX NUMBER (up to 7 Octal Digits)
- LEVEL - HEX LEVEL (13 - 6 used in MADEM)
- PUP - Pointer to parent. HEX block
- PDOWN - Pointer to daughter HEX block
- TERRAIN - At level 7 points to HEXELEV which contains altitude of the hex in meters
- PUOL - Pointer to unit occupancy list buffer block at Level 6
- PEEPER - Pointer to PEEPER LIST buffer block at Level 10.

b) HEXELEV Block

- ELEVAT - Elevation in meters. Stored as a real variable.

c) Buffer Block

PSTART - Pointer to first list block
NUMLINK - Number of links in list block chain.

d) Link Block

PNEXT - Pointer to next link block
PSB - Pointer to unit scoreboard

e. Linkages to other Data Structures

Link blocks point to the unit scoreboards (SB) of units which occupy a hex in the UOL list and units which can be seen from a hex in the PEEPER LIST.

f. Notes

- It appears that in the unMIDASized version of the model the BUFFER BLOCK is not used. However, it was included in this documentation because it exists in the MIDAS tables.
- UOL and PL's may exist at all levels in the hex tree.

E. COMMAND/CONTROL STRUCTURES

The command/control structures are used to simulate the command and control functions associated with the NATO and PACT command hierarchies. The three structures in this category include the PLAYER's LISTS, C2 TREES, and the PASSIVE TARGET LIST. The players lists are used to access particular unit types. They are used extensively by the semantic processing routines to assemble the red and blue C2 TREES. The C2 TREES represent the hierarchies of red and blue players in the model. They are the core of the command/control simulation. The PASSIVE TARGET LIST is a list of blue units that are not part of the C2 simulation, but merely targets for red attacks. These structures are explained in detail in the following subsections.

1. Player Lists

a. Data Block Index

PLAYERBUFFER

PLYST

C2

b. Description

The red and blue PLAYER LISTS are used to access COMMAND/CONTROL blocks (C2) of specified types. This structure allows the semantic processing routines to access units without traversing the C2 TREE. This capability is used to construct the C2 TREE in response to VOIL input.

The PLAYER LIST structures consist of a buffer word (PLAYERBUFFER) which points to an array (PLYST) within ISPACE which is dimensioned to the largest number of units in the scenario. Each word in the PLYST array contains a unit type code and a pointer to a C2 block of that type.

c. Structure Overview

1) Structure Diagram (Figure III-4)

2) Block Definitions

PLAYERBUFFER

Player List Buffer.

PLYST

Player List Array. Contains unit type codes and corresponding pointers to C2 blocks

C2

Command/Control Block. Contains pointers for C2 Tree. Along with unit number, side and type.

d. Block Specifications

1) Block Diagrams

a) PLAYERBUFFER

PTRPL	VARWORD
-------	---------

b) PLYST

ILW	IRW
ILW	IRW

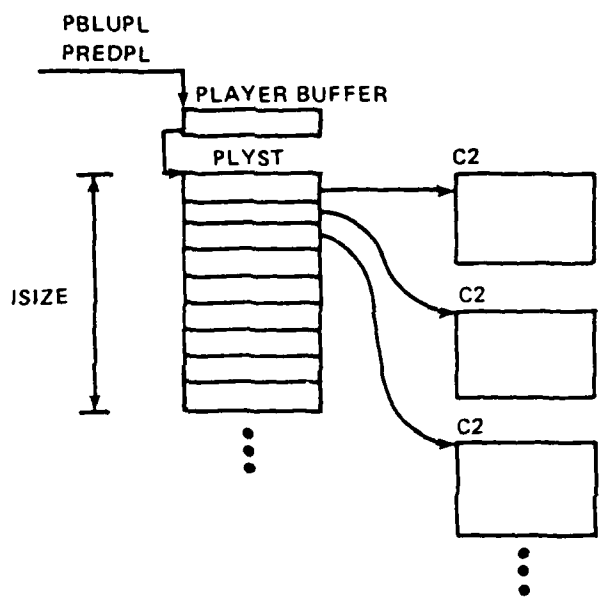


Figure III-4. Player List Structure Diagram

c) C2

UNIT NUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

2) Field Definitions

a) PLAYERBUFFER Block

PTRPL Pointer to PLYST Player List
VARWORD Total number of players in the list

b) PLYST Array

ILW Unit type code (See subsection f)
IRW Pointer to the unit's C2 block.

c) C2 Block

UNITNUMBER Number of the unit. If negative the unit is a passive target.

PUP Pointer to the C2 Block of the unit's commander.

PDOWN Pointer to the C2 block of the unit's subordinate.

PNEXT Pointer to the C2 block of the unit's sibling.

PSB Pointer to the SB block of the unit.
UNITTYPE the unit's type code. (See subsection f)

SIDE Unit Affiliation.
 1 = Blue (NATO)
 2 = Red (PACT)

e. Linkages to Other Data Structures

The player lists are linked via the C2 blocks to the C2 Tree and Passive Target Lists.

f. Notes

The UNITYTYPE Code Definitions used in MADEM are listed in
Appendix G.

MADAM UNITTYPE CODE DEFINITIONS

Name	<u>Decimal Number</u>	<u>Octal Number</u>
HAWK BTRY	170	252
HGRC BTRY	180	264
PAT BTRY	175	257
HAWK BOC	150	226
HERC BOC	160	240
PAT BOC	155	233
CRC	130	202
AIR BASE	220	334
TAB	158	236
AWACS	132	204
LANCE	86	126
HJ	34	42
BRIDGE	39	47
DEPOS	40	50
PERSHING	83	123
POL	84	124
SASP	210	322
ASP	87	127
RESERVES	88	130
TRAINS	89	131
CLV BTRY	94	136
VII BTRY	90	132
CORP CP	95	137
DIV CP	92	134
SOC	153	231
ATAF	128	200

AIRCRAFT UNITTYPE CODES (INTERNAL)

	<u>Decimal</u>	<u>Octal</u>
<u>BLUE</u>	401 - 419	621 - 643
AWAC	499	763

	<u>Decimal</u>	<u>Octal</u>
<u>RED</u>		
Fighters	420 - 439	644 - 607
F/B	440 - 459	670 - 713
Bombers	460 - 478	714 - 737
Unassigned	480 - 498	704 - 762

2. C2 Trees

a. Data Block Index

C2

PLBUFFER

SB

SDB

b. Description

Two tree data structures are used to represent the command and control hierarchies for the Blue (NATO) and Red (PACT) forces. Only player, as opposed to passive target, units are included in these trees. The overall structure of these trees is shown in the structure diagram below. The command and control hierarchies for red and blue forces used by the model are shown in subsection f.

c. Structure Overview

1) Structure Diagram (Figure III-5)

2) Block Definitions

PLBUFFER - C2 Tree Buffer Block. Contains pointer to the tree and the side of the tree.

- Command/Control Block. Contains tree pointers, unit number, pointer to scoreboard (SB), unit type code and side.

SB - Scoreboard Block. Contains pointers to C2 and HEX blocks, and the status display board (SDB). Also contains pointers to acquisition devices various unit status blocks and the future event list. Use varies with unit type.

SDB - Status Display Board Block. Contains subordinate, acquisition and order pointers. Use varies with unit type.

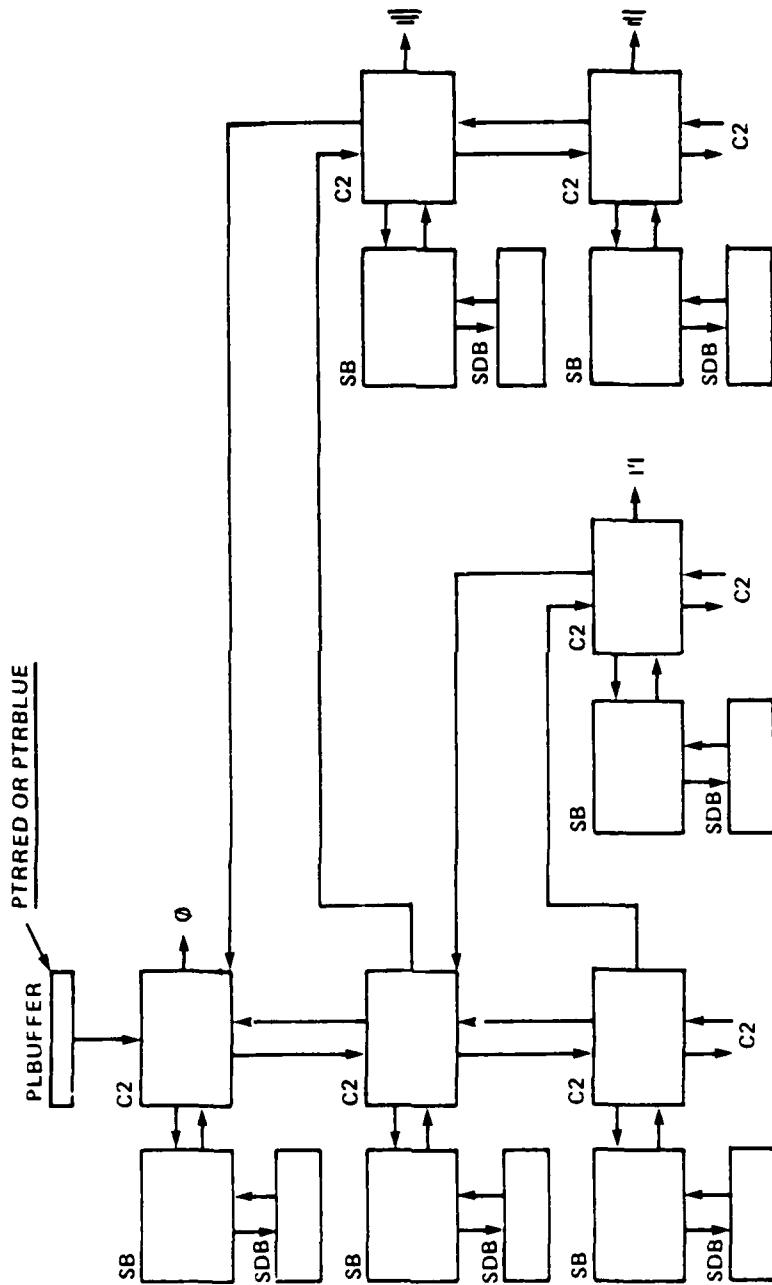


Figure III-5. C2 Tree Structure Diagram

4368/79W

d. Block Specifications

1) Block Diagrams

a) PLBUFFER

PTRPL	VARWORD
-------	---------

b) C2

UNIT NUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

Note: + Alternate Definition of Field

c) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATABASE	PABSTATNS
	+PARCFTSTAT
	+RBOCSTAT
	+PBTRYSTAT
	+STATUS

d) SDB

PSB	PSEEBUF
	+PSEE
SUBORDINATE	ORD
	+PRAID

Field Definitions

a) PLBUFFER Block

PTRPL Pointer to C2 block
 VARWORD Side Indicator for tree:
 = Blue (NATO)
 = Red (PACT)

b) C2 Block

- UNITNUMBER - Number of the unit. If negative the unit is a passive target.
- PUP - Pointer to the C2 block of the unit's commander
- PDOWN - Pointer to the C2 block of the unit's subordinate.
- PNEXT - Pointer to the C2 block of the unit sibling.
- PSB - Pointer to the SB block of the unit.
- UNITTYPE - The unit's type code (see subsection f)
- SIDE - Unit Affiliation.
 - 1 = Blue (NATO)
 - 2 = Red (PACT)

c) SB Block

- ADDRESS - Pointer to HEX block of the HEX in which the unit is located
- PC2 - Pointer to C2 block of the units.
- PSDB - Pointer to the SDB block of the unit.
- PFEL - Pointer to future event list event block.
- PACQ - Pointer to ACQBUF block. Used by CRC's for acquisition devices.
- ID - Identification number
- DATABASE - If BOC or BTRY points to ADSITEDG block
- PABSTATUS - Points to ABSTATUS block if the unit is an air base
- +PARCFTSTAT - Points to ARCFTSTATUS block if the unit is a flight of aircraft.

- +PBOCSTAT - Points to BOCSTAT block if the unit is a battalion operations center.
- +PBTRYSTAT - Points to BTRYSTAT block if the units is an antiaircraft battery.
- +STATUS - Alternative Field Definition.
- d) SDB Block
- PSB - Pointer to the unit's SB block.
- PSEEBUF - Pointer to SEEBUF block which is used by aircraft flight units to record damage levels. This field definition is used only by flights.
- +PSEE - Pointer to CRCEES block which is used by CRC units to record the blue and red flights it sees, and by the TRC as a pointer to its assigned targets list TARGETLISTBLOK.
- SUBORDINATE - Points to different types of subordinate description or target description blocks depending upon the unit type. Possible unit type vs. field use combinations are as follows:

<u>Unit Type</u>	<u>Subordinate Points to</u>
CRC	SUB
BOC	SUBLIST
BTRY	Fireunit
- ORD - Points to Orders block if the unit is a flight.

+PRAID - Points to the RAIDBLOK if the unit is the red theater commander.

e. Linkages to other Data Structures

The C2 TREES are linked to various unit status structures depending upon the type of unit represented by the C2 blocks.

f. Notes

Figures III-6, and III-7 show the hierarchies of blue and red player units. These hierarchies are representative of model processes only and are not direct models of the real world.

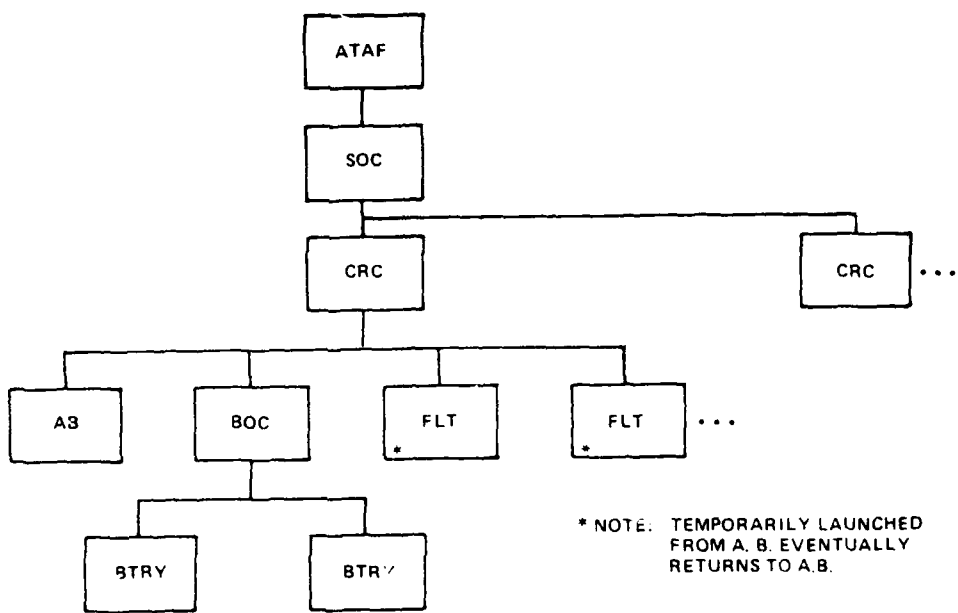


Figure III-6. Blue Command/Control Structure

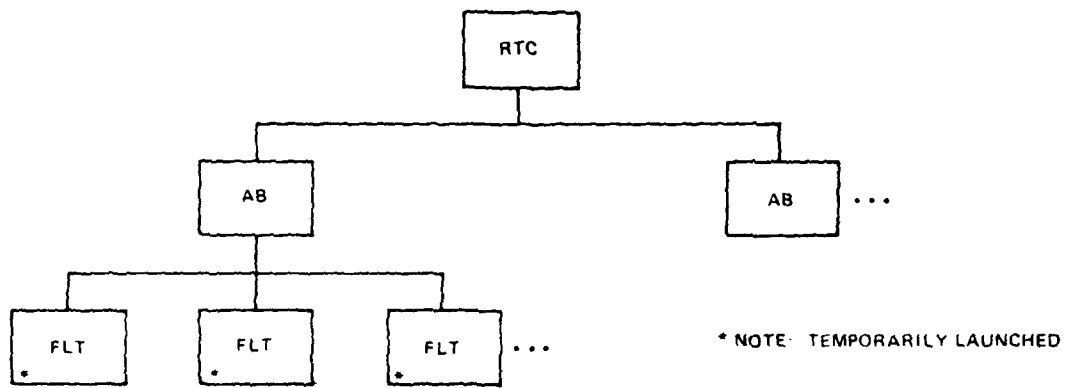


Figure III-7. Red Command/Control Structure

3. Passive Target List

a. Data Block Index

C2

SB (Modified)

b. Description

The PASSIVE TARGET LIST is made up of C2 and associated SB blocks which are in a linked list. This list is separate from the C2 TREE and is used to represent Blue units which are non-players. These Blue units are non-players in the sense that they merely act as objectives for red attacks.

Passive target units are characterized by negative unit numbers in their C2 blocks and a modified SB block which has three rather than the usual four words. The configuration of the passive target list is shown in the structure diagram.

c. Structure Overview

1) Structure Diagram (III-8)

2) Block Definitions

C2 - Command/Control Block. Contains list pointers, a negative unit number, pointer to SB, unit type code and side.

SB (Modified) - Scoreboard Block. Three rather than the usual four words. Contains pointers to C2 and HEX blocks.

d. Block Specifications

1) Block Diagrams

a) C2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

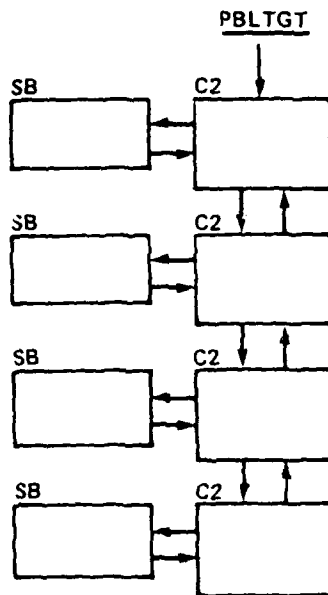


Figure III-8. Passive Target List Structure Diagram

4368/79W

b) SB (Modified)

ADDRESS	PC2
PSDB	PEEL
PACQ	ID

2) Field Definitions

a) C2 Block

- UNITNUMBER - Number of the Unit. If negative, the unit is a passive target.
- PUP - Pointer to the C2 block of the units commander.
- PDOWN - Pointer to the C2 block of the unit's sibling.
- PSB - Pointer to the SB block of the unit.
- UNITTYPE - The unit's type code (see subsection f)
- SIDE - Unit Affiliation.
1 = Blue (NATO)
2 = Red (PACT)
- PACQ - Pointer to ACQBUF block. Used by CRC's for acquisition devices (not used)
- ID - Identification Number

e. Linkages to Other Data Structures

The PASSIVE TARGET LIST is used exclusively for targeting by the red planning module. Its only external unrage is to the HEX block in which the unit is located.

f. Notes

F. RED STRUCTURES

The Red Structures are used to simulate the Red Threat Planning and attack processes. The Red Theater Commander (RTC) is the focal point of all other Red Structures. The RED THEATER COMMANDER structures together with its associated CORRIDOR DESCRIPTION lists and ATTACK REQUIREMENTS/ ALLOCATIONS lists controls the assignment of flights to specific targets. Potential targets are obtained from the POTENTIAL TARGET LIST and placed on either the ASSIGNED TARGET list or NONAVAILABLE TARGET list by the RTC. Orders are then generated for each flight which determine the mission profile for the COMMINS attack. Once flights have been launched many of the planning structures mentioned above are no longer required. They are therefore released for other uses.

1. Red Theater Commander

a. Data Block Index

C²

RAIDBLOK

SB

SDB

STDBLOK

TARGETLISTBLOK

TTDBLOK

WAVEBLOK

b. Description

The Red Theater Commander is the core of the Red Attack Planning Process. It controls execution of raids and waves through its RAIDBLOK/WAVEBLOK list. Similarly, choice of targets is controlled by the commander's PERCEPTIONS LIST (also referred to as the ASSIGNED TARGET LIST). The Red Theater Commander also controls the allocation of flights to targets on its PERCEPTIONS LIST through pointers from its WAVEBLOK blocks to the REQUIREMENTS/ALLOCATIONS LISTS. In addition, pointers from its RAIDBLOK blocks to CORRIDOR DESCRIPTION LISTS allow the Red Commander to Assign attacks to various corridors.

c. Structure Overview (Figure III-9)

1) Structure Diagram

2) Block Definitions

<u>C²</u>	-	<u>COMMAND/CONTROL BLOCK</u> . Contains unit no, type, side and C ² PTRS.
<u>SB</u>	-	<u>SCOREBOARD BLOCK</u> . Contains HEX PTR, Unit ID, PTR to FEL.
<u>SDB</u>	-	<u>STATUS DISPLAY BLOCK</u> . Contains PTRS to assigned tgt list (perceptions list) and RADIBLOK initialization.
<u>RAIDBLOK</u>	-	<u>RAID DESCRIPTION BLOCK</u> . Contains basic parameters for a raid including no. of waves and corridors.

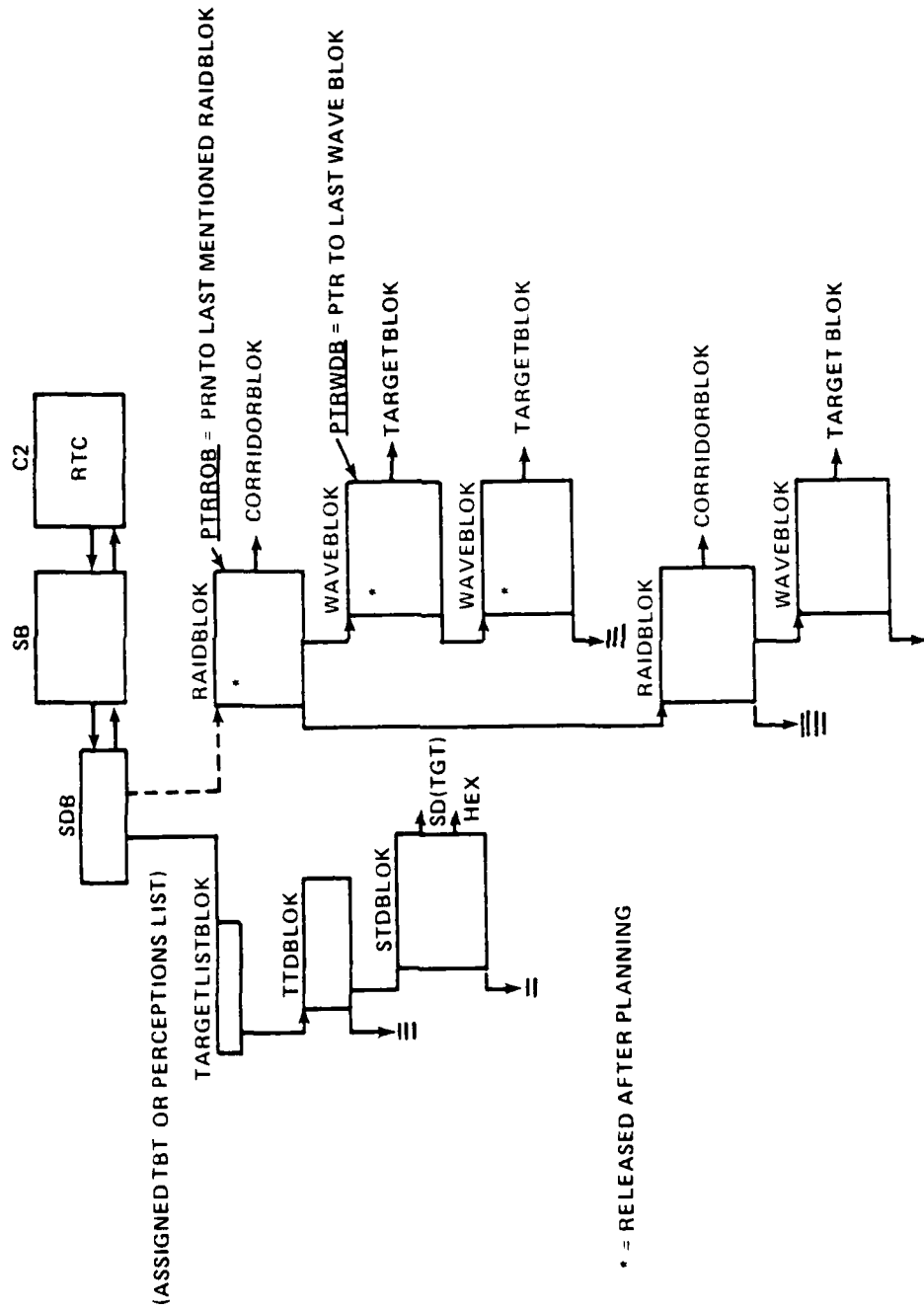


Figure III-9. Red Theater Commander Structure Diagram

436B/79W

- WAVEBLOK - WAVE DESCRIPTION BLOCK. Contains basic parameters for a wave including no. of tgt types, start time and duration.
- TARGETLISTBLOK - TARGET LIST BUFFER. Contains number of tgt type (TTDBLOK) blocks in the list
- TTDBLOK - TARGET TYPE BLOCK. Contains TGT type code and PTR to next TTDBLOK type, also contains PTR to a list of specific tgts of the same type (STDBLOK) and a count of the no. of specific targets.
- STDBLOK - SPECIFIC TARGET DESCRIPTION BLOCK. Contains PTRS to tft SB and HEX and to its TGTPTREE. Also contains perceived damage level and a PTR to the next STDBLOK.

d. Block Specifications

1) Block Diagrams

a) C^2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

b) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATA BASE	PABSTANS
	+ PARCPTSTAT
	+ PBOCSTAT
	+ PBTRYSTAT
	+ BTATUS

c) SDB

PSB	PSEEBUF
+	PSEE
SUBORDINATE	ORD
+	PRAID

d) TARGET LIST BLOK

PTRTYTL	NOTYTPL
---------	---------

d) TTDBLOK

PNEXT	NRTGTYP
PTGTPL	NOTGTPL

f) STDBLOK

PNEXT	PTGTSB
PTGTLTR	
DAMAGPER (SPACE)	
PADRPER	

g) RAIDBLOK

PNEXT	NRRAID
PTRWAVE	NDWAVES
PTALORD	NOCORDS

h) WAVEBLOK

PNEXT	NRWAVE
PTGTYPE	NOTGTYP
STARTIME (SPACE)	
DURATION (SPACE)	

2) Field Definitions

a) C² BLOCK

- UNITNUMBER - number of the unit.
- PUP - Pointer to the C² block of the unit's commander.
- PDOWN - Pointer to the C² block of the unit's subordinate.
- PNEXT - Pointer to the C² block of the unit's sibling.
- PSB - Pointer to the SB block of the unit.
- UNITTYPE - The Unit's type code.
- SIDE - Unit affiliation.
 - 1 = BLUE
 - 2 = RED (PACT)

b) SB Block

- ADDRESS - Pointer to HEX block of the hex in which the unit is located.
- PL2 - Pointer to C² block of the unit.
- PSDB - Pointer to the SDB block of the unit.
- PFEL - Pointer to future event list EVENT block.
- PACQ - Pointer to ACQBUF block. Used by CRC's for acquisition devices.
- ID - Identification number.
- DATABASE - If BOC or BTRY points to ADSITEDB block.
- PBSTATUS - Points to ABSTATUS block if the unit is an airbase.

- + PARCFTSTAT - Points to AFCFTSTATUS block if the unit is a flight of aircraft.
 - + PBOCSTAT - Points to BOCSTAT block if the unit is a battalion operations center.
 - + PBTRYSTAT - Points to BTRYSTAT block if the unit is an anti-aircraft battery.
 - + STATUS - Alternative Field Definition.
- c) SDB Block
- PSB - Pointer to the Unit's SB block.
 - PSEEBUF - Pointer to SEEBUF block which is used by aircraft flight units to record targets seen and their perceived damage levels. This field definition is used only by flights.
 - + PSEE - Pointer to CRCEES block which is used by CRC units to record the blue and red flights it sees. Also used by the RTC to point to its assigned targets list TARGETLISTBLOK.
-
- SUBORDINATE - Points to different types of subordinate description or target description blocks depending upon the unit type. Possible unit type vs. field use combinations are as follows:

	<u>UNIT TYPE</u>	<u>SUBORDINATE POINTS TO</u>
	CRC	SUB
	BOC	SUBLIST
	BTRY	FIREUNIT
ORD	-	Points to ORDERS block if the unit is a flight.
+ PRAID	-	Points to the RAIDBLOK if the unit is the red theater commander.
d)	<u>TARGETLISTBLOK BLOCK</u>	
	PRTYTL	- Pointer to TTDBLOK block
	NOTYTPL	- Number of TTDBLOK blocks (target types) in the list.
e)	<u>TTDBLOK BLOCK</u>	
	PNEXT	- Pointer to next TTDBLOK block in the list
	NRTGTYP	- Pointer to the STDBLOK block list
	PTGTPL	- Pointer to the STDBLOK block list.
	NGTGTPL	- Number of STDBLOK blocks in the list.
f)	<u>STDBLOK BLOCK</u>	
	PNEXT	- Pointer to the next STDBLOK block in the ASSIGNED TARGET LIST.
	PTGTSB	- Pointer to SB block of the specific target.
	PTGTLTR	- Pointer to TGTPTREE node
	DAMAGPER	- Perceived damage to target. Stored as a real variable. Initialized to 2.

- PADRPER - Pointer to HEX block in which target unit is located.
- g) RAIDBL BLOCK
- PNEXT - Pointer to the next RAIDBLOK block in the RAIDBLOK list.
 - NRRAID - Raid number.
 - PTRWAVE - Pointer to WAVEBLOK BLOCK. First in list.
 - NOWAVES - Number of waves in the list.
 - PTRCORD - Pointer to first CORRIDORBLOK in the CORRIDOR DESCRIPTION LIST.
 - NOCORDS - Number of corridors.
- h) WAVEBLOK BLOCK
- PNEXT - Pointer to the next WAVEBLOK block in the list.
 - NRWAVE - Wave number
 - PTGTYP - Pointer to first TARGETBLOK block in the list of target types to be attacked in the wave.
 - NOTGTYP - Number of target types in the wave.
 - STARTIME - Start time. Real variable
 - DURATION - Length of wave. Real variable.

e. Linkages to Other Data Structures

The Red Theater Commander shares its STDBLOK blocks with the POTENTIAL TARGET LIST and the NONAVAILABLE TARGET LIST. In addition, its RAIDBLOK blocks point to the CORRIDOR DESCRIPTION STRUCTURES and its WAVEBLOK blocks point to the attack REQUIREMENTS/ALLOCATION structures.

f. Notes

2. Potential Target List

a. Data Block Index

FOREST
FORTGTBUFFER
STDBLOK
TGPTREE

b. Description

The POTENTIAL TARGET LIST is used to keep track of potential targets by the Red Theater Planning Module. It consists of a linker list of FOREST blocks. Each of which corresponds to a particular target type found in the Blue C² TREE and PASSIVE TARGET LIST (See D.2 and D.3). Each FOREST block points to a leftist tree composed of linked TGPTREE and STDBLOK blocks. These trees contain location, damage and command/control information for specific targets of the type found in the origin FOREST blocks. These specific target trees are sometimes referred to as PROBABALISTIC EVENT TREES.

Within each tree the TGPTREE blocks act as the nodes of the tree while the STDBLOK blocks act as repositories of key information on the target. The nodes of the tree are sorted by damage level with the least damaged target residing at the top of the tree. For a complete explanation of the leftist tree sorting algorithm see APPENDIX E. Perceived damage is stored in STDBLOK while actual damage is stored in TGPTREE block.

STDBLOK blocks may be strung into linked lists to form the assigned target list discussed in E.2. The overall configuration of the POTENTIAL TARGET LIST is shown in the structure diagram.

c. Structure Overview

1) Structure Diagram (Figure III-10)

2) Block Definitions

FORTGTBUFFER	-	FOREST LIST BUFFER BLOCK. Contains number of forest blocks.
FOREST	-	TARGET TYPE BLOCK. Contains TGTYPE code and a PTR to a tree of specific targets of that type.

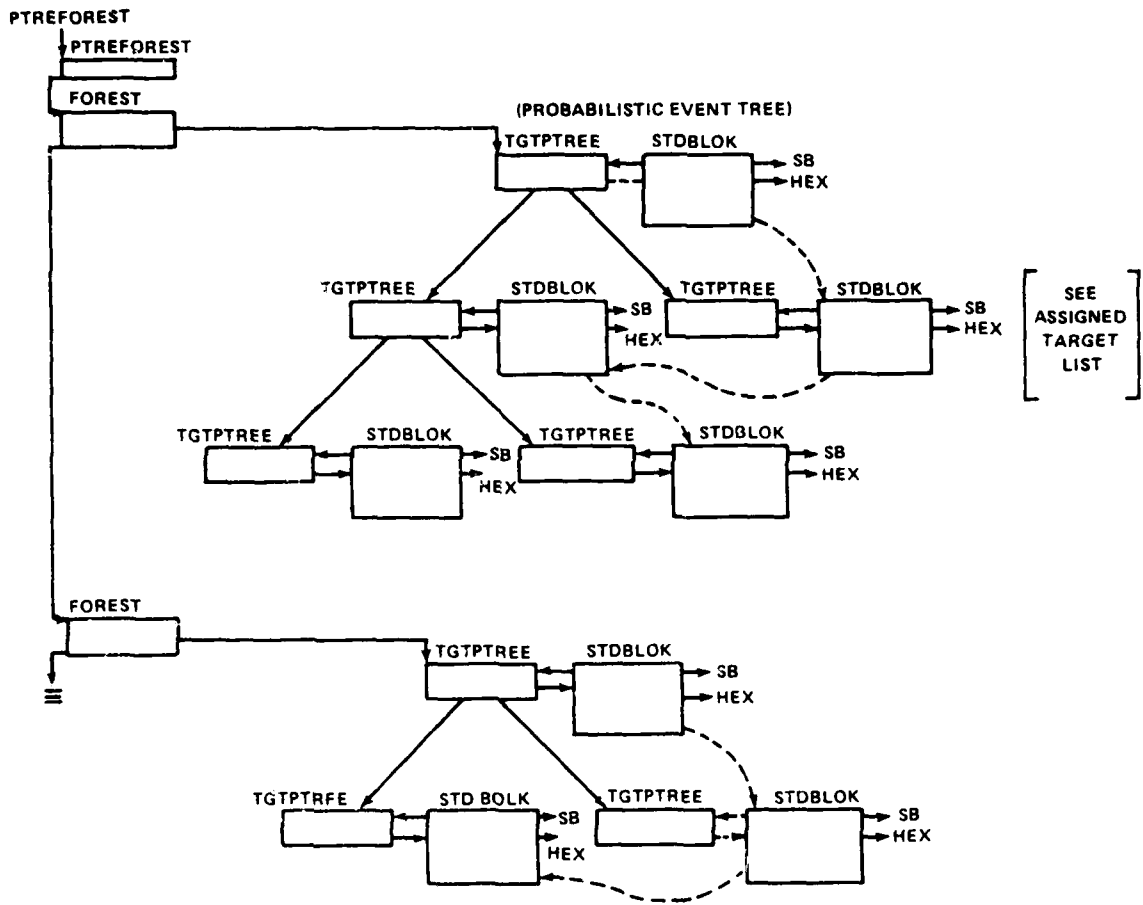


Figure III-10. Potential Target List Structure Diagram

- TGPTREE - TARGET TREE NODE. Contains PTAS to left and right nodes of leftist tree used to sort specific TGTS by damage level. Also contains PTR to its corresponding STDBLOCK. TGPTREE damage level is the actual damage level.
- STDBLOK - SPECIFIC TARGET DESCRIPTION BLOCK. Contains PTRS to TGT SB and HEX and to its corresponding TFTPTREE. Also contains the perceived damage level for the TGT. STDBLOCKS have a PNEXT field which allows them to be threaded into a linked list of assigned targets.

d. Block Specifications

1) Block Diagrams

a) FORTGTBUFFER

PFOREST	VARWORD
---------	---------

b) FOREST

PNEXT	NRTYPE
PTREE	

c) TGPTREE

DAMAGE			
PSTDBLOK	DIST	PLEPT	PRITE

d) STDBLOK

PNEXT	PTGTSB
PTGTLTR	
DAMAGPER	
PADRP	F-9

2) Field Definitions

a) FORTFTBUFFER BLOCK

PFOREST - Pointer to FOREST block.
VARWORD - Not used

b) FOREST BLOCK

PNEXT - Pointer to next FOREST block.
PTREE - Pointer to top TGTPTREE block in probabilistic event tree.

c) TGTPTREE NODE

DAMAGE - Actual damage to the target. Stored as a real variable. Initialized to 2.0
PSTDBLOK - Pointer to STDBLOK block.
DIST - Distance leaf of the tree in nodes.
PLEFT - Pointer to left TGTPTREE node in the tree
PRITE - Pointer to right TGTPTREE node in the tree.

d) STDBLOK BLOCK

PNEXT - Pointer to the next STDBLOK block in the ASSIGNED TARGET LIST.
PTGTSB - Pointer to SB block of the specific target.
PTGTLTR - Pointer to TGTpTREE node
DAMAGPER - Perceived damage to target. Stored as a real variable. Initialized to 2.
PAORPER - Pointer to HEX block in which target unit is located.

e. Linkages to other Data Structures

The POTENTIAL TARGET LIST shares its STDBLOK blocks with the ASSIGNED TARGET LIST and the NONAVAILABLE TARGET LIST.

f. Notes

The relationship of the potential and assigned target lists is illustrated in Figure III-11.

3. Assigned Target List
(Red Commander Perceptions List)

a. Data Block Index

ATTACKBLOK
SDB
STDBLOK
TARGETLISTBLOK
TTDBLOK

b. Description

The ASSIGNED TARGET LIST is used by the Red Theater Planning module to keep track of targets designated for attack in the current raid. It is actually a series of nested lists with lists of specific target description blocks (STDBLOK) stratified by target type. The STDBLOK blocks are obtained from the POTENTIAL TARGET LIST. The desired STDBLOK blocks in the potential target list are pointed to be specific target resource allocation blocks (ATTACKBLOK) created by the Red Theater Planner when resources are matched to attack requirements. (See E.4 for further details). The overall configuration of the ASSIGNED TARGET LIST is shown in the structure diagram. The assigned target list is attached to the Red Theater Commander's SDB block.

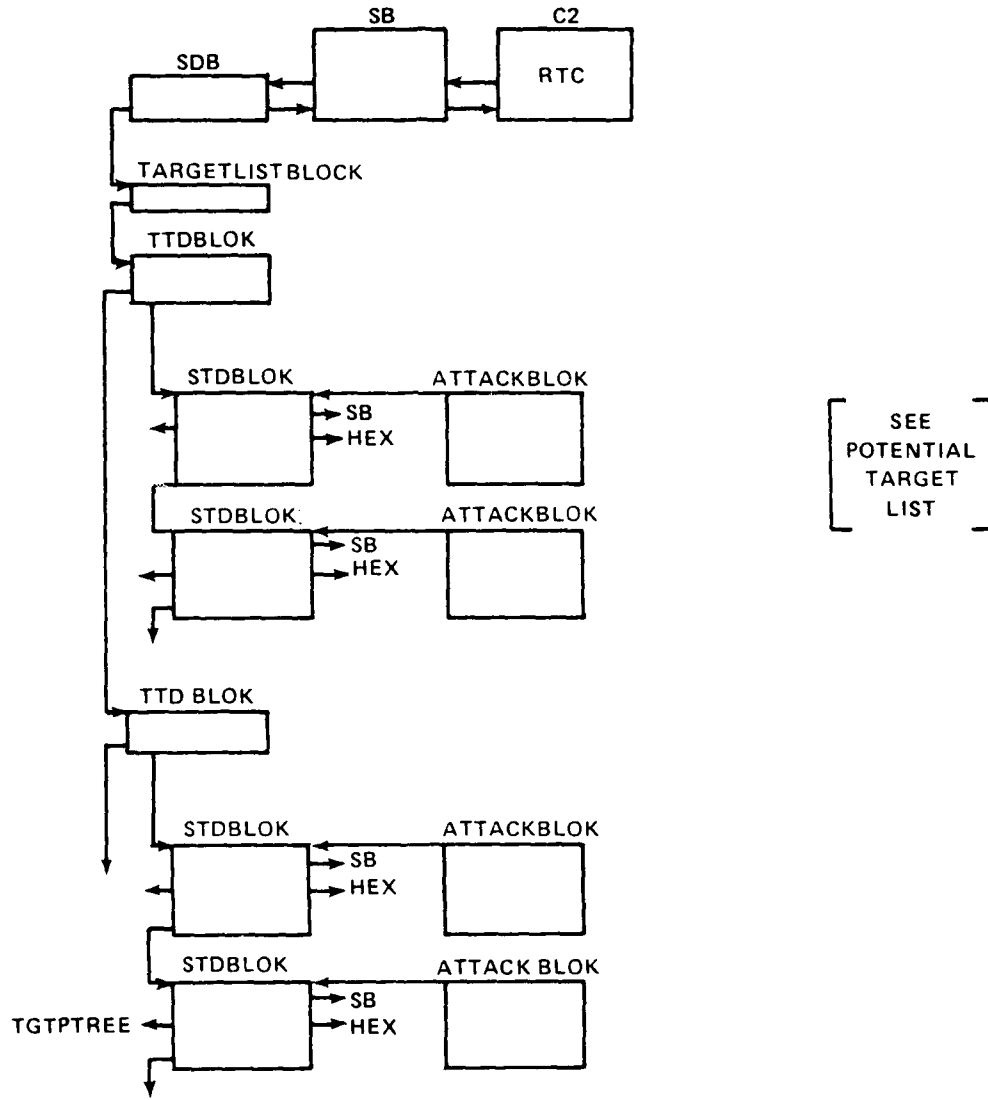
c. Structure Overview

1) Structure Diagram (Figure III-12)

2) Block Definitions

TARGETLISTBLOK - TARGET LIST BUFFER. Contains number of TGT TYPE (TTDBLOK) blocks in the list.

TTDBLOK - TARGET TYPE BLOCK. Contains TGT type code and PTR to next TTDBLOK type, also contains PTR to 4 list of specific tgts. of the same type (STDBLOK) and a count of the no. of specific targets.



4368/79W

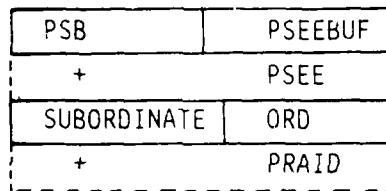
Figure III-12. Assigned Target List Structure Diagram (Red Commander Perception List)

- STDBLOK - SPECIFIC TARGET DESCRIPTION BLOCK.
Contains PTRS to tgt. SB and HEX and to its TGTPTREE. Also contains perceived damage level and a PTR to the next STDBLOK
- SDB - STATUS DISPLAY BOARD of Red Commander
- SB - SCOREBOARD of Red Commander
- C² - COMMAND CONTROL BLOCK of Red Commander
- ATTACKBLOK - SPECIFIC TARGET ATTACK RESOURCE ALLOCATION BLOCK for target.

d. Block Specifications

1. Block Diagrams

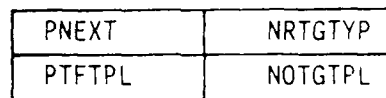
a) SDB



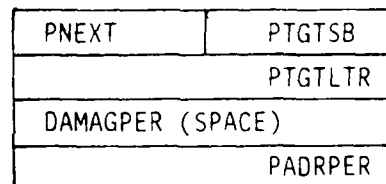
b) TARGETLISTBLOK



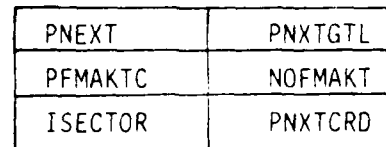
c) TTDBLOK



d) STDBLOK



e) ATTACKBLOK



2) Field Definitions

a) SBD BLOCK

- PSB - Pointer to the Unit's SB block.
- PSEEBUF - Pointer to SEEBUF block which is used by aircraft flight units to record targets seen and their perceived damage levels. This field definition is used only by flights.
- +PSEE - Pointer to CRCEES block which is used by CRC units to record the Blue and Red Flights it sees. Also used by the RTC as a pointer to its assigned targets list. TARGETLISTBLOK.
- SUBORDINATE - Points to different types of subordinate description or target description blocks depending upon the unit type. Possible unit type vs. field use combinations are as follows:

<u>UNIT TYPE</u>	<u>SUBORDINATE POINTS TO</u>
CRC	SUB
BOC	SUBLIST
BTRY	FIREUNIT

- ORD - Points to ORDERS block if the unit is a flight.
- + PRAID - Points to the RAIDBLOK if the unit is the Red Theater commander.

b) TARGETLISTBLOK BLOCK

- PTRYTTL - Pointer to TTDBLOK block
- NOTYTPL - Number of TTDBLOK blocks (target types) in the list.

c) TTDBLOK BLOCK

- PNEXT - Pointer to next TTDBLK block in the list
- NRTGTYP - Target type code.
- PTGTPL - Pointer to the STDBLOK block list
- NOTGTPL - Number of STDBLOK blocks in the list

d) STDBLOK BLOCK

- PNEXT - Pointer to the next STDBLOK block in the ASSIGNED TARGET LIST.
- PTGTSB - Pointer to SB block of the specific target.
- PTGTLTR - Pointer to TFTPREE node.
- DAMAGPER - Perceived damage to target. Stored as a real variable. Initialized to 2.
- PADRPER - Pointer to HEX block in which target unit is located.

e) ATTACK BLOCK

- PNEXT - Pointer to next ATTACK BLOK in list.
- PNXTGTL - Pointer to STDBLOK block.
- PFAKTG - Pointer to FAKTGBLOK block.
- NOFMAKT - Number of FAKTGBLOK blocks (formation attacking target) in the list.
- ISECTOR - Sector of the attack relative to the attack corridor. (see corridor)
- PNXTCRD - Pointer to the CORRIDOR block for the attack.

e. Linkages to Other Data Structures

The ASSIGNED TARGET LIST shares its STDBLOK blocks with the POTENTIAL TARGET LIST and the NONAVAILABLE TARGET LIST. In addition its STDBLOK blocks are pointed to by ATTACKBLOK blocks which reside in the attack REQUIREMENTS/ALLOCATIONS structure.

f. Notes

Figure III-11 illustrates the relationship of the potential and assigned target lists. This relationship is crucial to an understanding of the target assignment process.

4. Nonavailable Target List

a. Data Block Index

NOAVAILBLOK

STDBLOK

b. Description

The NONAVAILABLE TARGET LIST is used by the Red Theater Planning Module to keep track of targets which cannot be attacked in the current raid. Targets are placed on the NONAVAILABLE TARGET LIST when they are not in the allowable attack shown in Subsection F and when attack resources are not sufficient to match mission requirements. The overall configuration of the nonavailable target list is shown in the structure diagram.

c. Structure Overview

1, Structure Diagram (Figure III-13)

2) Block Definition

NOAVAILBLOK - NON-AVAILABLE TARGET BLOCK. Created by TGTGONE routine and used by other targets which are not available for attack due to lack of resources or geographic unsuitability. Contains PTR to STDBLOCK and PTR to next NOAVAILBLOK. Also contains perceived damage level. (Projected damage level).

STDBLOK - SPECIFIC TARGET DESCRIPTION BLOCK. Contains target SB and HEX PTRS as well as a PTR to the TGTPTREE. Also contains the perceived damage level for the target and a PNEXT field which allows them to be threaded into a linked list of assigned targets.

PNOAVAL = PTR IN COMMOM THTRPLN

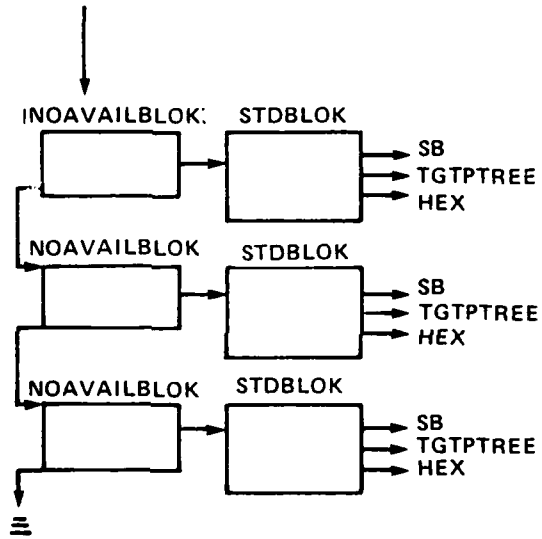


Figure III-13. Nonavailable Target List Structure Diagram

4368/79W

d. Block Specifications

1) Block Diagrams

a) NOAVAILBLOK

PNEXT
PJDAMAGE
PSTDBLOK

b) STDBLOK

PNEXT	PTGTSB
PTGTLTR	
DAMAGPER	
PADRPER	

2) Field Definitions

a) NOAVAILBLOK BLOCK

- PNEXT - Pointer to NOAVAILBLOK block
- PJDAMAGE - Projected damage level.
- PSTDBLOK - Pointer to STDBLOK block on the POTENTIAL TARGET LIST.

b) STDBLOK BLOCK

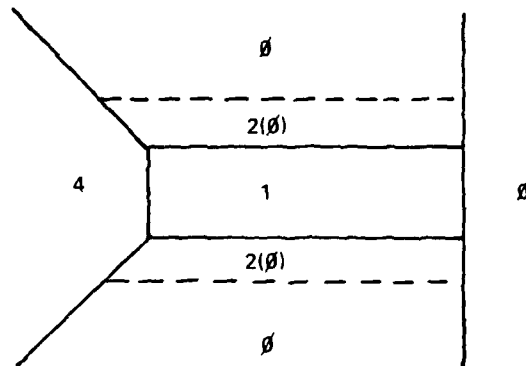
- PNEXT - Pointer to the next STDBLOK block in the ASSIGNED TARGET LIST.
- PTGTSB - Pointer to SB block of the specific target.
- PTGTLTR - Pointer to TGTPTREE node
- DAMAGPER - Perceived damage to target. Stored as a real variable. Initialized to 2.
- PADRPER - Pointer to HEX block in which target unit is located.

e. Linkages to Other Data Structures

The nonavailable target list shares its' STDBLOK blocks with the POTENTIAL TARGET LIST and the ASSIGNED TARGET LIST.

f. Notes

Figure III-14 illustrates the attack sectors relative to an attack corridor.



ACCEPTABLE
TARGET
AREAS

ATTACK ZONE = 4
CORRIDOR CENTER = 1
RIGHT BUFFER = 3
LEFT BUFFER = 2
UNACCEPTABLE = ø

Figure III-14. Attack Sectors

5. Corridor Description Lists

a. Data Block Index

ABVCR
ACTAB
CONSTBLOK
CORRIDORBLOK
HEXBLOK
HEXLINK

b. Description

The CORRIDOR DESCRIPTION LISTS are used by the Red Theater Planning Module to specify the boundaries of the attack corridors input by the user. They are also used to keep track of the assignment of airbases to corridors within aircraft range and the types of aircraft available on the bases. The general configuration of attack corridors is illustrated in Subsection F.

c. Structure Overview

1) Structure Diagram (Figure III-15)

2) Block Definitions

<u>CORRIDORBLOK</u>	-	<u>CORRIDOR DESCRIPTION BLOCK</u> . Contains basic corridor parameters input by user.
<u>CONSTBLOK</u>	-	<u>CORRIDOR CONSTANTS BLOCK</u> . Contains values calculated from corridorblok contents which describe the corridor boundaries.
<u>HEXBLOK</u>	-	<u>HEXLINK BUFFER BLOCK</u> . Contains part to five HEXLINK lists which specify the corridor in terms of hexes.
<u>HEXLINK</u>	-	<u>HEXLINK BLOCK</u> . Contains ptrs. to next HEXLINK and PTR to HEX.

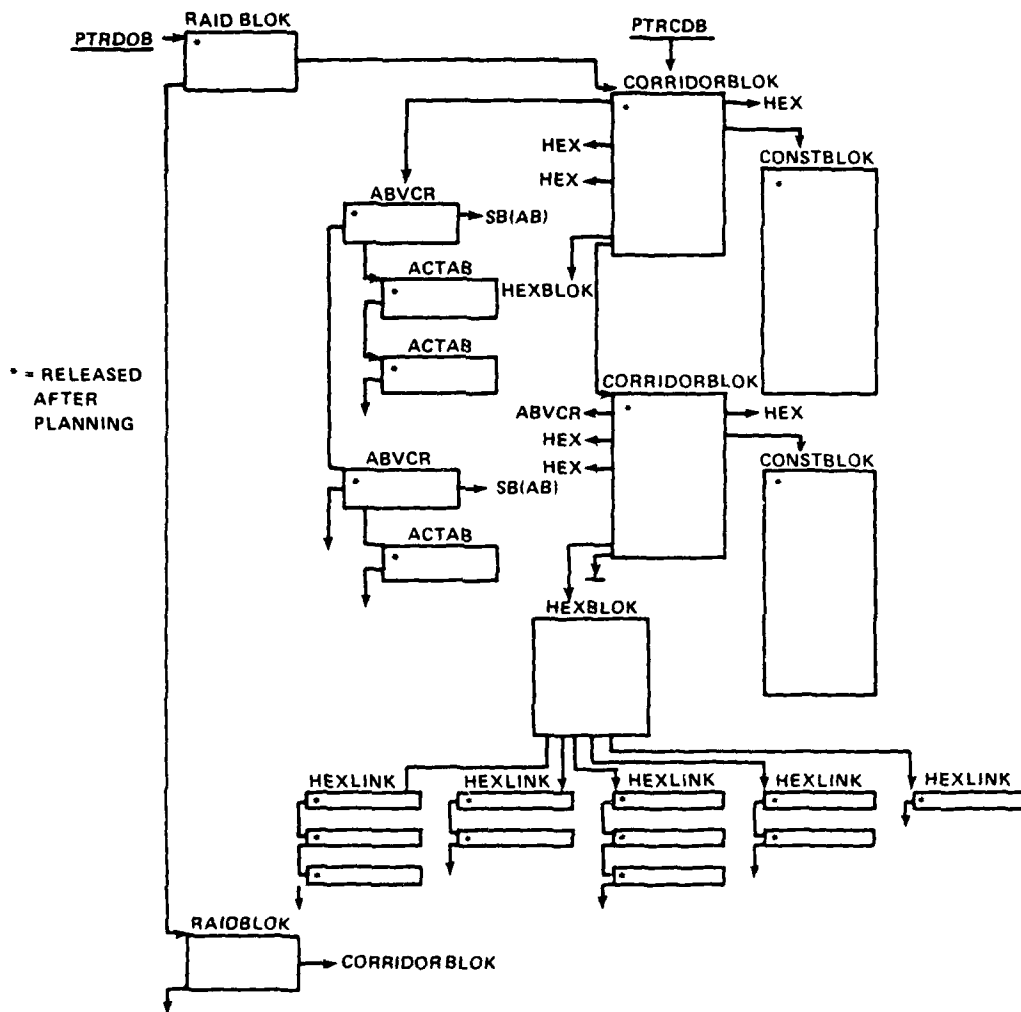


Figure III-15. Corridor Description List Structure Diagram

4368/79W

- ABVCR - AIR BASE VS. CORRIDOR BLOCK. Air vs. corridor assignment contains PTR to next ABVCR, PTR to air base SB, the number of aircraft on the air base, and PTR to ACTAB list.
- ACTAB - AIRCRAFT ON AIRBASE BLOCK. Aircraft assignment to airbase. Contains a/c type, no on hand at base, no assigned, and PTR to next ACTAB.

d. Block Specifications

1) Block Diagrams

a) CORRIDORBLOK

PNEXT	NRCORD
PABUSCOR	NOABVCR
PLHEX	PRHEX
PCHEx	NHWIDTH
PHLIST	PBDCNST
DEPTHLR (SPACE)	
ANGCORD (SPACE)	
ANGSPRD (SPACE)	
BUFRWDH (SPACE)	

b) CONSTBLOK

CORDScope
YINTLBUF
YINTLCOR
YINTRCOR
YINTRBUF
XSPREAD
YSPREAD
ENTRYSLOPE
YLINENTRY
YENDCORD

c) HEXBLOK

ITOTAL	
PHEXY	NOLIST1
PHEX1	NOLIST2
PHEX3	NOLIST3
PHEX6	NOLIST4
PHEXUNK	NOLISTUNK

d) HEXLINK

PNEXT	HEX
-------	-----

e) ABVCR

PNEXT	PTRABSB
PACTAB	NOACTAB

f) ACTAB

PNEXT	NRACTYP
NOALOH	NOACASN

2) Field Definitions

a) CORRIDORBLOK BLOK

- PNEXT - Pointer to next CORRIDORBLOK block
- NRCORD - Corridor Number.
- PLHEX - Pointer to left corridor hex.
- PRHEX - Pointer to right corridor hex.
- PCHEx - Pointer to centerline corridor hex.
- PBDCNST - Pointer to CONSTBLOK block.
- PABVSCOR - Pointer to ABVCR block list.
- NOABVCT - Number of ABVER blocks in the list.
- PHLIST - Pointer to HEXBLOK block.
- DEPTHCR - Depth of corridor, real variable.
- ANGCORD - Angle of corridor, real azimuth
- ANGSPRD - Spread angle of corridor exit. real
- BUFRWDH - Sam buffer zone width. real

b) CONSTBLOK BLOK

- CORDSCLOPE - Corridor slope
- YINTLBUF - Y intercept of buffer zone boundary-left.

YINTRCOR	-	Y intercept of corridor boundary-right.
YINTRBUF	-	Y intercept of buffer zone boundary - right
XSPREAD	-	X coordinate of exit spread lines.
YSPREAD	-	Y coordinate of exit spread lines.
ENTRYSLOPE	-	Slope of corridor center line.
YLINENTRY	-	Y intercept of corridor center line.
YENDCORD	-	Y intercept of corridor exit.
c) <u>HEXBLOK BLOCK</u>		
ITOTAL	-	
PHEXY	-	Pointer to HEXLINK list (corridor left to corridor center)
NOLIST1	-	Number of (PHEX4) HEXLINK blocks in the list
PHEX1	-	Pointer to HEXLINK list (buffer left to corridor left)
NOLIST2	-	Number of (PHEX1) HEXLINK blocks in the list.
PHEX3	-	Pointer to HEXLINK list (corridor right to buffer right)
NOLIST3	-	Number of (PHEX3) HEXLINK blocks in the list.
PHEX6	-	Pointer to HEXLINK list (corridor center to corridor right)
NOLIST4	-	Number of (PHEX6) HEXLINK blocks in the list.
PHEXUNK	-	Number of HEXLINK list hexes between corridor ends)
NOLISTUNK	-	Number of (PHEXUNK) HEXLINK blocks in the list.

d) HEXLINK BLOCK

- PNEXT - Pointer to next HEXLINK block in the list.
- HEX - Pointer to HEX block (Note: could also be a nex number)

e) ABVCR BLOCK

- PNEXT - Pointer to next ABVCR block in the list
- PTRABSB - Pointer to airbase SB block
- PACTAB - Pointer to ACTAB block
- NOACTAB - Number of ACTAB blocks in the list

f) ACTAB BLOCK

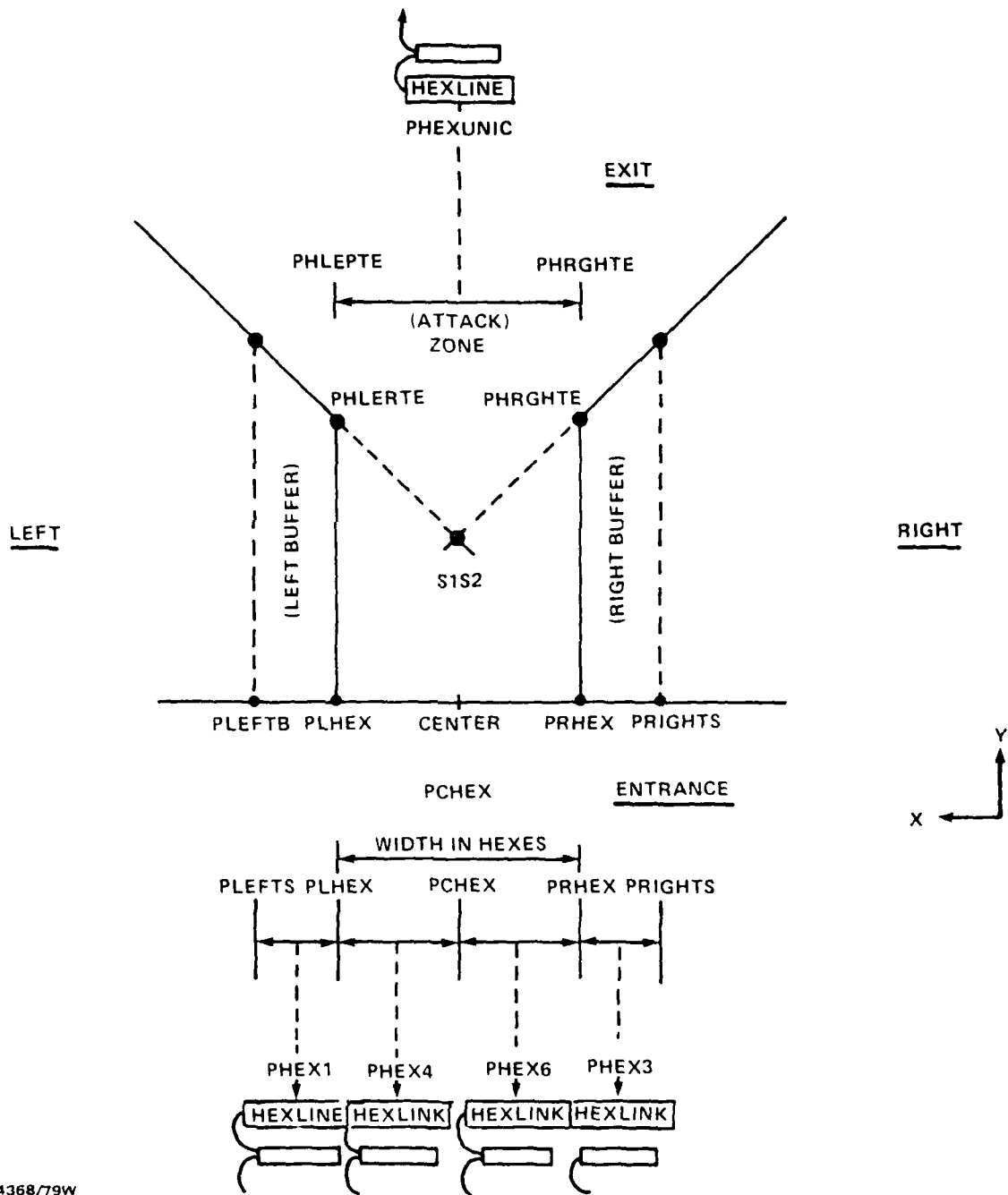
- PNEXT - Pointer to next ACTAB in the list
- NRACTYP - Aircraft type code
- NOACOH - Number of aircraft on hand
- NOACASN - Number of aircraft assigned

e. Linkages to other Data Structures

The CORRIDOR DESCRIPTION LISTS are pointed to by a RAIDBLOK block which describes the raid in which the corridor is used.

f. Notes

Figure III-16 illustrates the relationship of HEXLINK lists to corridor boundaries.



4368/79W

Figure III-16. Relationship of HEX LINK Lists To Corridor Boundaries

6. Attack Requirements/Allocations Lists

a. Data Block Index

ATTACKBLOK

ACTAB

COMMAND

FAKTGBLOK

FDBDBLOK

FLTAKTBLOK

FMFLTDB

FORMATIONBLOK

b. Description

The REQUIREMENTS LIST is used by the Red Theater Planner to specify the number and composition of formations required to attack targets of various types. Each TARGETBLOK block corresponds to a generic target type (e.g., airbase, boc, btry). FORMATIONBLOK blocks correspond to the formation types required for a target type. Each FORMATIONBLOK points to a FDBDBLOK block (formation data base) which in turn points to a list of FMFLTDB blocks (formation flight description) which point to FLTDB blocks (flight data base) for various types of flights. Each flight is homogeneous with respect to aircraft type. The aircraft for a flight must be assembled on a single air base. Partial flights cannot be launched. However, formations can be assembled from flights originating at multiple air bases. The formations required for each target type must be specified by the user for each raid via the UOL.

The ALLOCATIONS LIST parallels the REQUIREMENTS LIST. It corresponds to the actual allocation of aircraft resources to specific targets of the types specified in the TARGETBLOK blocks. The structure diagram illustrates this parallel structure. FKTGBLOK blocks (formation attacking target) in the ALLOCATIONS LIST correspond to the FORMATIONBLOK blocks in the REQUIREMENTS list. Similarly, FLTAKTBLOK blocks (flight attacking target) correspond to FMFLTDB blocks. It should be noted that all blocks indicated by an asterisk (*) are released after planning, the

only blocks which remain after planning are the database and command blocks required for subsequent operation of FLIGHTS created by the planning module.

c. Structure Overview

1) Structure Diagram (Figure III-17)

2) Block Definitions

REQUIREMENTS LIST DEFINITIONS - resource requirements for target types.

WAVEBLOK - WAVE DESCRIPTION BLOCK. Contains PTRS to next wave and TARGETBLOK list along with the wave number and the number of TARGETBLOK types in the list also contains the start time and duration of the wave.

TARGETBLOK - TARGET TYPE RESOURCE REQUIREMENTS BLOCK. Contains target type code, PTRS to formation and attackblok lists and the number of formations and attackbloks in each list. Also contains the maximum aircraft allocation for the TGT type, the number allocated, and the minimum and maximum attack radius.

FORMATIONBLOK - FORMATION DESCRIPTION BLOCK. Contains PTRS to next FORMATION BLOCK and to FDBDBLOK as well as the number of formations required and assigned

FDBDBLOK - FORMATION DATA BASE BLOCK. Contains basic formation type specifications including pointer to flight data base list (FMFLTDB).

FMFLTDB - FORMATION COMPONENT FLIGHT DATA BLOCK. Contains PTR to FLTDB flight data base block for a flight attached to the formation.

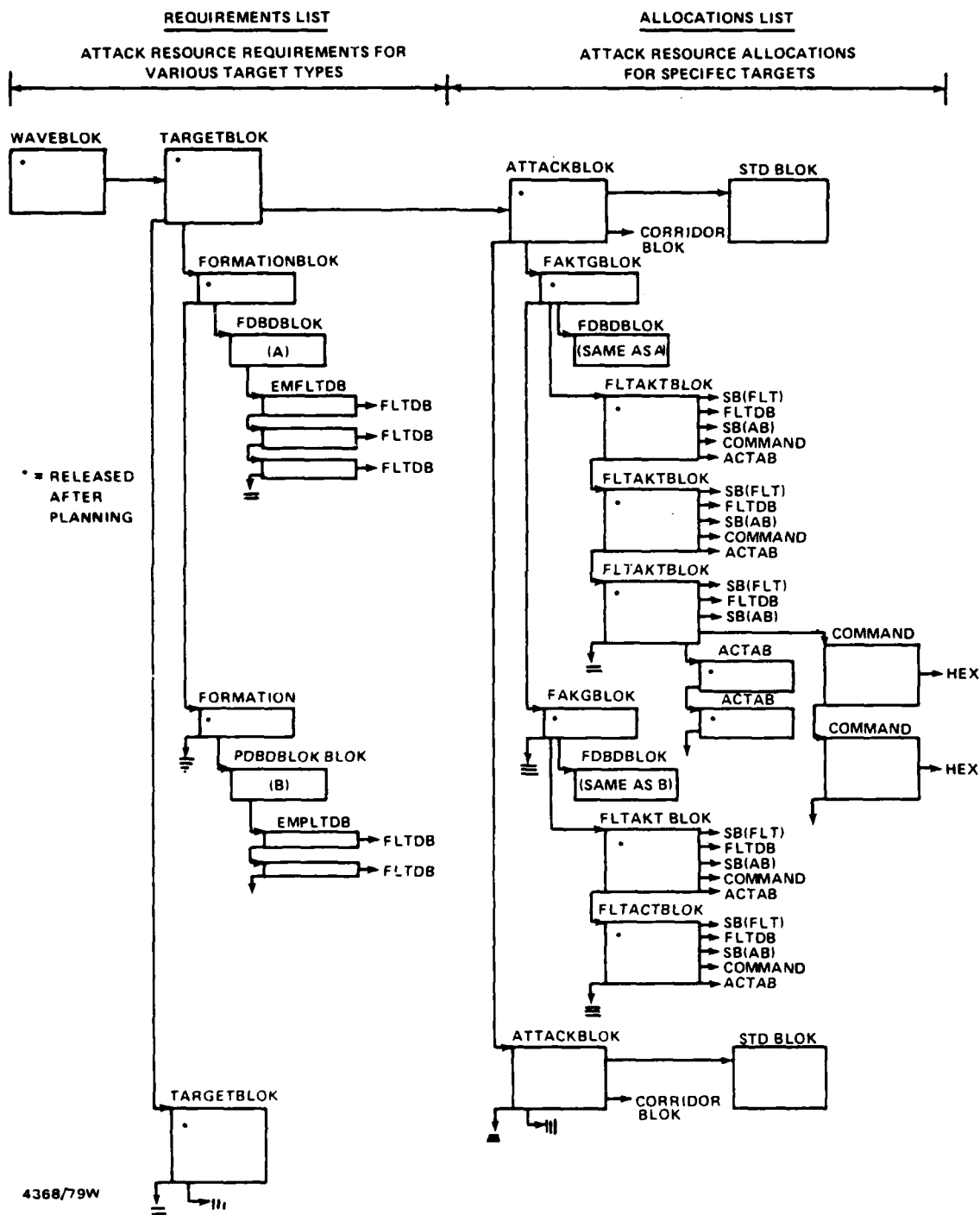


Figure III-17. Attack Requirements/Allocations List Structure Diagram

- ALLOCATIONS LIST DEFINITIONS - Resource Allocations for specific targets.
- ATTACKBLOK - SPECIFIC TARGET RESOURCE ALLOCATION BLOCK. Contains PTRS to specific target description block (STDBLOK), formations attacking targetBlist (FAKTGBLOK) and the CORRIDOR block for the corridor through which the attack will take place. Also contains attack sector.
- FAKTGBLOK - FORMATIONS ATTACKING TARGET BLOCK. Contains description of a formation assigned to attack the target. Contains PTRS to the FDBDBLOK described above, the next FAKTGBLOK attacking the TGT and a list of FLTAKTBLOK (flights attacking the TGT) which make up the formation. Also contains the number of FLTAKTBLOKS.
- FLTAKTBLOK - FLIGHTS ATTACKING TARGET BLOCK. Basic description of the flight, its composition, home base and orders.
- ACTAB - AIRCRAFT ON AIRBASE BLOCK. Number of aircraft of the type specified in FLTAKTBLOK on hand and assigned at FLT's home base.
- COMMAND - COMMAND FOR ENTITY. A list of up to six COMMAND blocks describes all of the actions which must be taken by the Red Flight in the course of its mission. Contains the action to be taken, a PTR to the hex address, and optionally the time at which the action is to occur.

d. Block Specifications

1) Block Diagrams

a) WAVEBLOK

PNEXT	NRWAVE
PTGTYPE	NOTGTYP
STARTIME (SPACE)	
DURATION (SPACE)	

b) TARGETBLOK

PNEXT	NRTGTYP
PTRFORM	NOFORM
PTGTATK	NOTGTAK
MAXACAL	NOACALC
MAXRHEX	NONRHEX

c) FORMATIONBLOK

PNEXT	PNXFRDB
NOFRMRQ	NOFRMAL

d) FDBDBLOK

PNEXT	NRFORM
PTRFLT	NOFLT
SPFORMC	

e) FMFLTDB

PNEXT	PNXFLODB
-------	----------

f) ATTACKBLOK

PNEXT	PNXTGTL
PFMAKTC	NOFMAKT
ISECTOR	PNXTCRD

g) FAKTGBLOK

PNEXT	PNXFRDB
PFLTAKT	NOFLAKT

h) FLTAKTBLOK

PNEXT	PNXFLDB
NOACFLT	PFLABSB
PNXACAB	PTRFRAG
PFLTSB	

i) ACTAB

PNEXT	NRACTYP
NOACOH	NOACASN

j) COMMAND

PNEXT	NUMACTS
TMFLG	ADDRESS
TIME (SPACE)	
ACTION	

k) STDBLOK

PNEXT	PTGTSB
PTGTLTR	
PAMGPER (SPACE)	
PADRPER	

2) Field Definitions

a) WAVE Block

PNEXT pointer to next WAVE block
 NRWAVE number of waves
 PTGTYPE pointer to TARGETBLOK
 NOTGTYP number of target types
 STARTTIME start time for wave
 DURATION duration of wave

b) TARGET Block

PNEXT pointer to next target block
 NRTGTYP number of target types
 PTRFORM pointer to FORMATIONBLOK
 NOFORM number of formations
 PRGTATK pointer to ATTACKBLOK

- NOTGTAK - number of attack blocks
 - MAXACAL - maximum AC allocations
 - NOACALC - number of AC allocations
 - MAXRHEX - maximum range in hexes at which the target can be attacked
 - MINRHEX - minimum range in hexes at which the target can be attacked
- c) FORMATIONBLOK
- PNEXT - pointer to next FORMATIONBLOK
 - NOFRMRQ - number of formation req.
 - NOFRMAL - actual number of formations
- d) FDBDBLOK
- PNEXT - pointer to next FDBDBLOK
 - NRFORM - formation number, must be unique
 - PTRFLT - pointer to formation flight data block (FMFLTDB)
 - NOFLT - number of flights in the formation
 - SPFORMC - formation cruise speed in meters/seconds
- e) FMFLTDB
- PNEXT - pointer to next formation flight block (FMFLTDB)
 - PNXFLDB - pointer to flight data block (FLTDB)
- f) ATTACKBLOK
- PNEXT - pointer to next ATTACKBLOK
 - PNXTGTL - pointer to STDBLOK
 - PFAKTG - pointer to formation attacking target block (FAKTGBLOK)
 - NOFAKT - number of formation attack blocks
 - ISECTOR - number of sector to be attacked, relative to attack corridor
 - PNXTCRD - pointer to corridor block (CORRIDORBLOK)

- g) FAKTGBLOK
- PNEXT - pointer to next formation attacking target block (FAKTGBLOK)
 - PNXFRDB - pointer to formation data base block (FDBDBLOK)
 - PFLTAKT - pointer to flight attacking target block (FLTAKTBLOK)
 - NOFLAKT - number of flights attacking target blocks
- h) FLTAKTBLOK
- PNEXT - pointer to next FLTAKTBLOK
 - PNXFLDB - pointer to FLTDB
 - NOACFLT - number of actual flights
 - PFLABSB - pointer to air base scoreboard
 - PNXACAB - pointer to aircraft on air base block (ACTAB)
 - PTRFRAG - pointer to COMMAND block
 - PFLTSB - pointer to flight scoreboard
- i) ACTAB
- PNEXT - pointer to next ACTAB
 - NRACTYP - aircraft type number
 - NOACOH - number of aircraft on hand
 - NOACASN - number of aircraft assigned
- j) COMMAND
- PNEXT - pointer to next COMMAND block
 - NUMACTS - number of the command
 - TMFLG - time flag, if 1 a time is associated with the command
 - ADDRESS - pointer to hex block for the command
 - TIME - time command is to be performed
 - ACTION - command or action code

k) STD BLOK

PNEXT	pointer to next STDBLOK
PTGTSB	pointer to target scoreboard
PTGTLTR	pointer to TGTPTREE
DAMGPER	perceived damage level
PADRPER	pointer to hex address

e. Linkages to Other Data Structures

The REQUIREMENTS LIST is pointed to by the WAVEBLOK block which is associated with the RED THEATER COMMANDER. The ATTACKBLOK block in the ALLOCATIONS LIST points to an STDBLOK block (specific target description block) in the Red theater commander's ASSIGNED TARGET LIST (also known as the Red theater commander perceptions list). Since these STDBLOK blocks are also in the POTENTIAL AND UNAVAILABLE TARGET LISTS, the allocations list is also connected to both the POTENTIAL AND UNAVAILABLE TARGETS LISTS.

f. Notes

7. RED AIR BASES

a. Data Block Index

ABINFO
ABQUEDB
ABSTATUS
ACOB
C2
QUEUES
QUESTAT
SB

b. Description

RED AIR BASE STRUCTURES are designed to keep track of the number and type of aircraft on the AIRBASE as well as the status of each aircraft type in terms of launch capability. The overall configuration of this structure is shown in the structure diagram.

c. Structure Overview

1) Structure Diagram (See Figure III-18)

2) Block Definitions

<u>C2</u>	-	<u>COMMAND CONTROL BLOCK</u> , contains unit NO, type and side, and C2 PTRS.
<u>SB</u>	-	<u>SCOREBOARD BLOCK</u> , contains Hex address, MR, unit ID, PTR to future event list for the unit, and <u>ABSTATUS</u> PTR.
<u>SDB</u>	-	<u>STATUS DISPLAY BLOCK</u> . Not used by AIRBASE.
<u>ABSTATUS</u>	-	<u>AIR BASE STATUS BLOCK</u> , contains PTRS to the AIRBASE information list (ABINFO) and the QUEUES list for each aircraft type on base. Also contains no of A/C on base and no of types of A/C on base.
<u>ABINFO</u>	-	<u>AIRCRAFT ON BASE INFORMATION BLOCK</u> . Contains aircraft type, no on hand and no in each service queue, and a PTR to A/C type's <u>ACDB</u> .

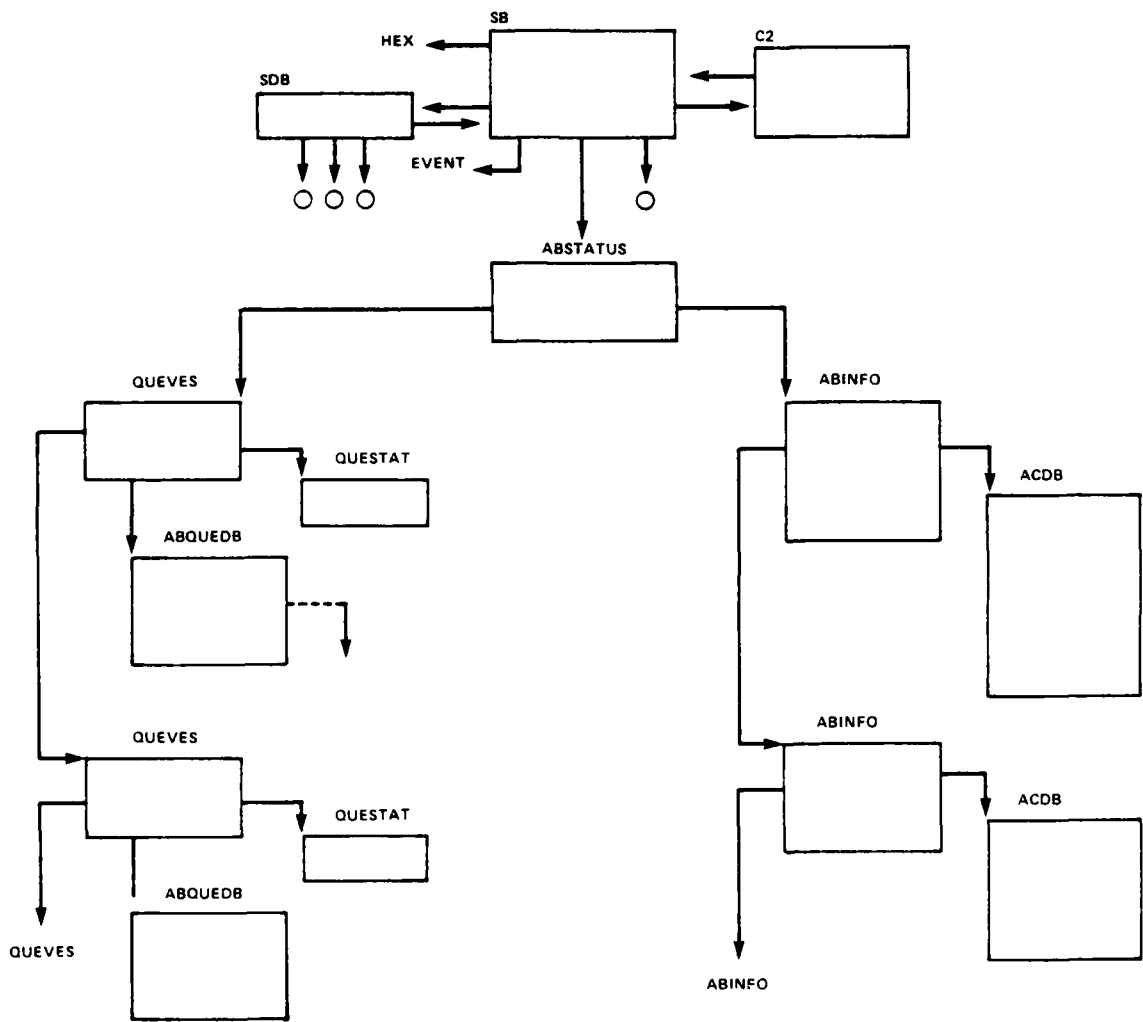


Figure III-18. Red Air Base Structure Diagram

4368/79W

- QUEUES - AIRCRAFT SERVICE QUEUE BLOCK.
- ABQUEDB - AIRBASE QUEUE DATA BASE BLOCK.
- QUESTAT - QUEUE STATUS BLOCK.
- ACDB - AIRCRAFT DATA BASE BLOCK. Operational parameters of A/C type.

d. Block Specification

1) Block Diagrams

a) C2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

b) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATABASE	PABSTATUS

c) ABSTATUS

PACTAB	NOACTAB
PTR2QUES	NOACONAB
ABDAMAGE (SPACE)	

d) QUEUES

NEXT	QUENUM
PTR	NUMBER
PQDB	PQUESTAT

e) QUESTAT

VALUE (SPACE)

f) ABQUEDB

PNEXT	CLASS
VALUE1(SPACE)	
VALUE2(SPACE)	
VALUE3(SPACE)	

g) ABINFO

NEXT	NRACTYP
NOACOH	PTRACDB
NORMRQ	
NOREARMQ	
NOREFVELQ	
NOLAVNCHQ	

h) ACDB

NEXT	NRACTYPE
MAXSPEED (SPACE)	
CRUISESPEED (SPACE)	
MAXALTITUDE (SPACE)	
MINALTITUDE (SPACE)	
MAXCLIMBDIVE (SPACE)	
FUELCONSUME (SPACE)	
ACQRANGE (SPACE)	
RADARCS (SPACE)	
ATTACKRADIUS (SPACE)	
MAXFUEL (SPACE)	

2) Field Specifications

a) C2 Block

UNITNUMBER unit number
PUP pointer to C2 block of unit's
commander
PDOWN pointer to C2 block of unit's
subordinate
PSB pointer unit's SB block
PNEXT pointer to unit's sibling. C2
block
UNITTYPE unit type code (220)
SIDE unit affiliation 2 = Red (Pact)

- b) SB Block
- | | |
|--------------------|--|
| ADDRESS | pointer to <u>HEX</u> block in which unit is located |
| PC2 | pointer to <u>C2</u> block of the unit |
| PSDB | pointer to SDB block (inactive) |
| PFEL | pointer to future event list for the unit |
| PACQ
(inactive) | pointer to acquisition devices |
| ID | unit ED number |
| DATABASE | pointer to data base block (inactive) |
- c) ABSTATUS Block
- | | |
|----------|--|
| PACTAB | pointer to <u>ABINFO</u> block |
| NOACTAB | number of <u>ABINFO</u> blocks in the list. Corresponds to number of aircraft on the base. |
| PTR2QUES | pointer to <u>QUEUES</u> block |
| NOACONAB | number of <u>QUEUES</u> blocks in the list. Corresponds to number of aircraft types on the base. |
| ABDAMAGE | damage level of base. Real variable. |
- d) QUEUES Block
- | | |
|---------------|---|
| NEXT | pointer to next <u>QUEUES</u> block in the list |
| QUENUM | <u>QUEUE</u> number. (2 = ready queue) |
| PTR
NUMBER | |
| PQDB | pointer to <u>ABQUEDB</u> block |
| QUESTAT | pointer to <u>QUESTAT</u> block |
- e) QUESTAT Block
- | | |
|-------|-------------------------|
| VALUE | unknown. Real variable. |
|-------|-------------------------|

f) ABQUEDB Block

PNEXT pointer to next ABQUEDB block
CLASS aircraft class
VALUE1
VALUE2 -
VALUE3 -

g) ABINFO Block

NEXT pointer to next ABINFO block
NRACTYP aircraft type code
NOACOH number of aircraft of type (NRACTYP)
on hand on the base
PTRACDB pointer to ACDB for the aircraft
type (NRACTYP)
NORMRQ number in repair queue
NOREARMQ number in rearm queue
NOREFUELQ number in refuel queue
NOLAUNCHQ number in launch queue

h) ACDB Block

NEXT pointer to next ACDB block in data
base (not used in this context)
NRACTYPE aircraft type code
MAXSPEED maximum speed. Real variable
CRUISESPEED cruising speed. Real variable
MAXALTITUDE maximum altitude. Real variable
MAXCLIMBDIVE maximum rate of altitude change.
Real variable
FUELCONSUME Fuel consumption rate. Real variable
ACQRANGE acquisition range. Real variable
RADARCS radar cross-section. Real variable
ATTACKRADIUS - maximum attack range of aircraft.
Real
variable
MAXFUEL maximum fuel capacity. Real variable

e. Linkages to Other Data Structures

f. Notes

QUEUES blocks and their related lists are not used at present.

8. Red Flights

a. Data Block Index

ACDB
AQDB
ARCFSAW
ARCFTSTATUS
C2
COMMAND
FLTDB
FORMATION
LOAD
MUN
ORDERES
PAYDBLOK
PAYLOAD
SB
SDB
SEEBUF
WINGMAN

b. Description

The Red Flight Data Structures control the actions of Red Flights. In addition to the three command/control blocks C², SB and SDB, Red Flights also use three lists. These lists include: the PERCEPTIONS LIST, the ORDERS LIST and the FLIGHT STATUS LIST.

The PERCEPTIONS LIST is composed of a buffer and a singly-linked list of ARCFSAW blocks. These ARCFSAW blocks contain information on Blue Targets perceived by the Red Flight. This information includes the location of the Blue Targets and its perceived damage level. When the Red Flight returns to its' airbase the perceived damage to Blue Target is transferred to the Red Theater Planner's STDBLOK blocks. Thus damage perception by Red Flights are transmitted to the Red Commander for subsequent planning activities.

The ORDERS LIST is composed of two buffered lists. The first is made-up of up to six COMMAND blocks which specify the actions to be taken by the flight at various points in its mission. These COMMAND blocks determine the flight geometry and mission profile for the flight. The second list is made up of WINGMAN blocks which contain pointers to the SR blocks of other flights in the other Red Flights in the formation.

The FLIGHT STATUS LIST consists of an ARCFTSTATUS block which tracks flight status and a set of two MUNITIONS LISTS which keep track of air-to-air and air-to-ground ordnance carried by the Red flight. The ARCFTSTATUS block points to an FLTOB block which is the core of a FLIGHT DATA BASE STRUCTURE. This structure is used as a template for construction of flights of specified types. It provides the basic aircraft characteristics and initial payload levels used to create and operate the FLIGHT.

Both the FLIGHT DATA STRUCTURES and the FLIGHT DATA BASE STRUCTURES are shown in the structure diagrams.

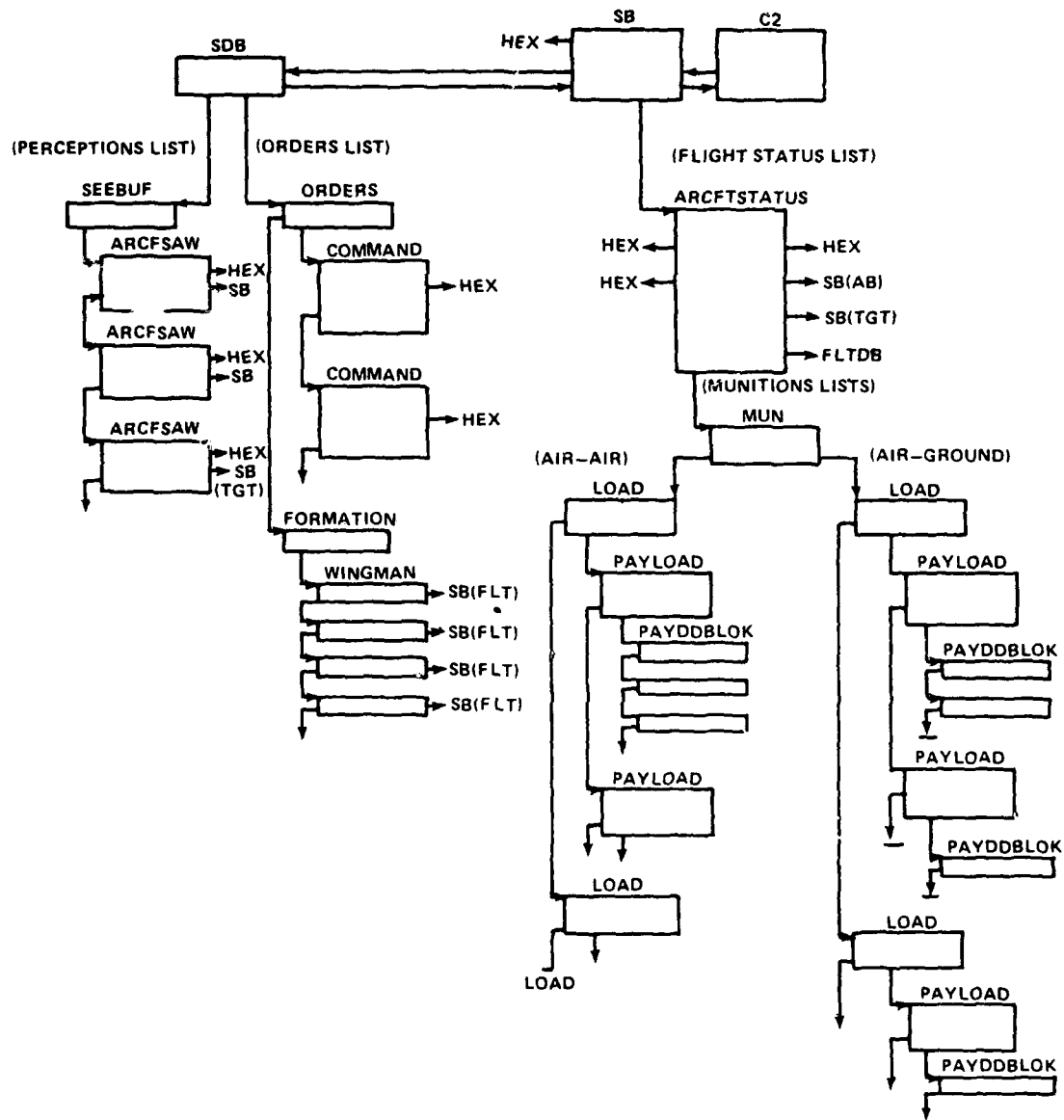
c. Structure Overview

1) Structure Diagrams (Figure III-19 & III-20)

2) Block Definitions

a) COMMAND/CONTROL BLOCKS:

- | | | |
|----------------------|---|--|
| <u>C²</u> | - | <u>COMMAND/CONTROL BLOCK</u> . Contains unit no, type, side, and C ² PTRS. |
| <u>SB</u> | - | <u>SCOREBOARD BLOCK</u> . Contains hex address, unit id, PTR to FEL and STAT. |
| <u>SDB</u> | - | <u>STATUS DISPLAY BLOCK</u> . Contains PTRS to perceptions list (SEEBUF) and orders list (ORDERS). Subordinate PTR not used by Red FLTS. |



4368/79W

Figure III-19. Red Flight Structure Diagram

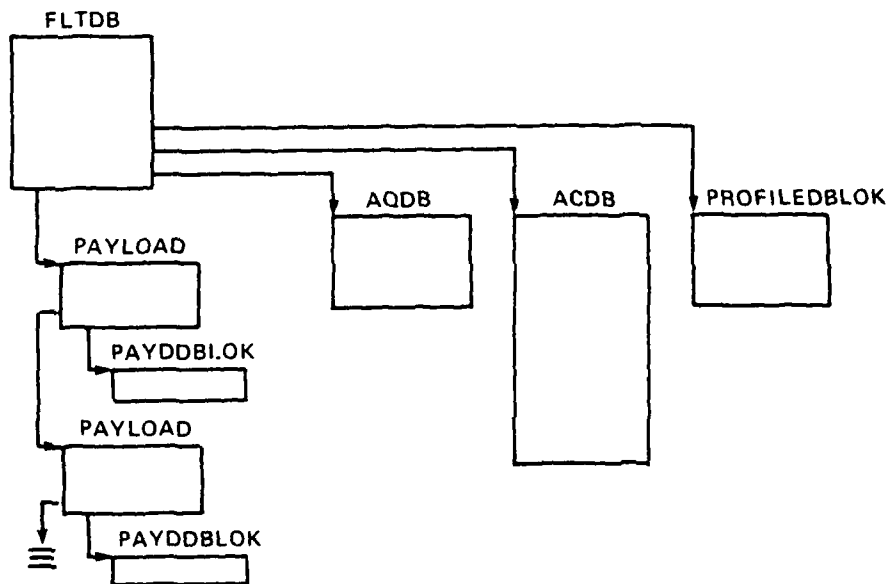


Figure III-20. Flight Data Base Structure Diagram

b) FLIGHT STATUS LIST

ARCFTSTATUS - FLIGHT STATUS BLOCK. Contains basic flight status information includes PTRS to starting, ending and next hex address in current move. Also includes PTRS to home base and TGT SB's, a status word, and current fuel, altitude, speed, and direction.

MUN - MUNITIONS LISTS BUFFER BLOCK. Buffer for air-to-air and air-to-ground munitions lists

PAYLOAD - PAYLOAD CLASS DESCRIPTION BLOCK. Basic payload parameters for class of ordinance. Contains class type, max and min amounts and max fire range. Also includes PTR to a list of attached ordnance types of the same class.

PAYDDBLOCK - PAYLOAD TYPE DATA BASE BLOCK. Contains PTR to next PAYDDBLOK and type of ordnance.

LOAD - PAYLOAD LIST BUFFER. Used to break payloads into types and keep track of ammunition load weight.

c) PERCEPTIONS LIST

SEEBUF - PERCEPTIONS LIST BUFFER BLOCK

ARCFSAW - AIRCRAFT PERCEPTION BLOCK. Contains information on entity perceived by a flight. Includes PTR to hex and SB of entity.

- d) ORDERS LIST
- ORDERS - ORDERS LIST BUFFER BLOCK.
Contains no of orders remaining
 - COMMMAND - COMMAND DESCRIPTION BLOCK.
Describes command to be followed
by flight at specific address.
Up to six in the list
 - FORMATION - FORMATION BUFFER BLOCK.
Contains number of flts in the
formation to which the flight
belongs.
 - WINGMAN - WINGMAN LIST BLOCK. Contains
PTR to other flts in the
formation
- 3) FLIGHT DATA BASE
- FLTDB - FLIGHT DATA BASE BLOCK.
Contains basic flight description
including No. of payloads,
maximum no of A/C, minimum no.
of A/C, and Multac, SPFLTC
(SPACE, DISTSER(SPACE))
 - PAYLOAD - PAYLOAD DESCRIPTION BLOCK.
Contains payload class (NRPDGLS)
max and min amount of payload,
and max fire range
 - PAYLDDBLOK - PAYLOAD TYPE BLOCK. Contains
payload type index
 - AQDB - ACQUISITION DEVICE DATA BASE BLOCK.
Contains type index and range
of aquisition device.
 - ACDB - AIRCRAFT DATA BASE BLOCK.
Contains A/C characteristics
such as speed, max range, etc.

PROFILEDBLOK - MISSION PROFILE BLOCK. Contains flight altitude levels for three phases of mission - Alt. to corridor entrance, alt to tgt, and alt from tgt to air base.

d. Block Specifications

1) Block Diagrams

a) C^2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

b) SB

ADDRESS	PC^2
PACQ	ID
DATABASE	PABSTATUS
	+ PARCFTSTAT
	+ PBOCSTAT
	+ STATUS

c) SDB

PSB	PSEEBUF
	+ PSEE
SUBORDINATE	ORD
	+ PRAID

d) SEEBUF

PTRSEE	NUNITS
--------	--------

e) ARCFSAW

PNEXT	PSB
ADDRESS	TYPE
DAMAGE (SPACE)	

f) ORDERS

PTRFORMS	PTRACT
----------	--------

g) COMMAND

PNEXT	NVMACTS
TMFLG	ADDRESS
TIME (SPACE)	
ACTION	

h) FORMATION

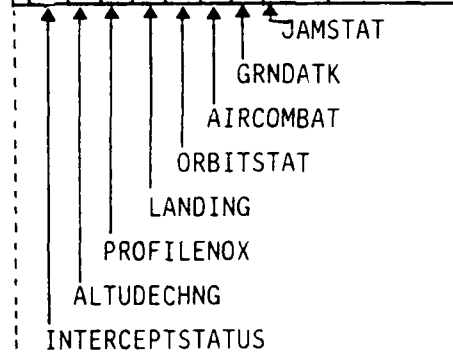
PFORM	NUMFLTS
-------	---------

i) WINGMAN

PNEXT	PSB
-------	-----

j) AFCFTSTATUS

PFLTDB	PNUMITIONS
PSTRTHX	PENDHX
PNXTHX	PAIRBASE
CNTRLMODE	PAIRTGT
PGNOTDG	NUMAIRCRAFT
	DUMMY



FLITELEG

FUEL (SPACE)
ALTITUDE (SPACE)
SPEED (SPACE)
DIRECTION (SPACE)

k) MUN

PAG	NUMAG
PAA	NUMAA

l) LOAD

PNEXT	TYPE
AMOUNT	PORDDDB

m) FLTDB

PNXFLDB	NRFLITE
PTYPLDS	NOPLYDS
PTYAQDB	PTACDB
MAXNOAC	MINNOAC
MULTAC	PROFILE
SPFLTC (SPACE)	
DISTSEP (SPACE)	

n) PAYLOAD

PNXTYPD	NRPDCLS
MAXAMT	MINAMT
MAXFIRE RANGE PAYLDDDB	

o) PAYLLDBLOK

NEXT	TYPEINDEX
------	-----------

p) AQDB

NEXT	NRAQTYP
RANGE (SPACE)	
NOUSE 1	
NOUSE 2	

q) PROFILEDBLOK

PNXPRDB	NRPROFL
ALTCREN (SPACE)	
ALTOAB (SPACE)	

r) ACDB

NEXT	NRACTYPE
MAXSPEED (SPACE)	
CRUISESPEED (SPACE)	
MAXALTITUDE (SPACE)	
MINALTITUDE (SPACE)	

MAXCLIMBDIVE (SPACE)
FUELCONSUME (SPACE)
ACQRANGE (SPACE)
RADARCS (SPACE)
ATTACKRADIUS (SPACE)
MAXFUEL (SPACE)

2) Field Definitions

a) C² Block

- UNITNUMBER - Number of the unit.
- PUP - Pointer to the C² block of the unit's commander.
- PDOWN - Pointer to the C² block of the unit's subordinate.
- PNEXT - Pointer to the C² block of the unit's sibling.
- PSB - Pointer to the SB block of the unit.
- UNITTYPE - The unit's type code (128)
- SIDE - Unit Affiliation. 2 - Red (PACT)

b) SB Block

- ADDRESS - Pointer to HEX block of the HEX in which the unit is located.
- PL2 - Pointer to C² block of the unit.
- PSDB - Pointer to the SDB block of the unit.
- PFEL - Pointer to future event list EVENT block.
- PACQ - Pointer to ACQBUF block. Used by CRC's for acquisition devices.

- ID - Identification number
- DATABASE - If BOC or BTRY points to ADSITEDB block.
- PABSTATUS - Points to ABSTATUS block if the unit is an air base.
- +PARCFTSTAT - Points to ARCFSTATUS block if the unit is a flight of aircraft.
- +PDOCSSTAT - Points to BOCSTAT block if the unit is a battalion operations center.
- +PBTRYSTAT - Points to BTRYSTAT block if the unit is an antiaircraft battery.
- +STATUS - Alternative field definition.

c) SDB Block

- PSB - Pointer to the unit's SB block.
- PSEEBUF - Pointer to SEEBUF block which is used only by flights.
- +PSEE - Pointer to CRCEES block which is used by CRC units to record the blue and red flights it sees. Also used by the BIC as a pointer to its assigned target block. TARGETLISTBLOK.
- SUBORDINATE - Points to different types of subordinate description or target description blocks depending upon the unit type. Possible unit type vs. field use combination are as follows:

<u>UNIT TYPE</u>	<u>SUBORDINATE POINTS TO</u>
CRC	SUB
BOC	SUBLIST
BTRY	FIREUNIT

- ORD - Points to ORDERS block if the unit is a flight.

- +PRAID - Points to the RAIDBLOK if the unit is the Red Theater Commander.
- d) SEEBUF BLOCK
- PTRSEE - Pointer to ARCFSAW block. (first in list)
- NUNITS - Number of ARCFSAW blocks in the list. Corresponds to number of targets perceived.
- e) ARCFSAW BLOCK
- PNEXT - Pointer to next ARCFSAW block in the list.
- PSB - Pointer to SB block of target.
- ADDRESS - Pointer to HEX block in which target is located.
- TYPE - Unit type code of target.
- DAMAGE - Perceived damage level of the target. Real variable.
- f) ORDERS BLOCK
- PTRFORMS - Pointer to FORMATION block.
- PTRACT - Pointer to COMMAND block.
- g) COMMAND BLOCK
- PNEXT - Pointer to next command block
- NUMACTS - Number of the command in the list.
- TMFLG - Time flag. If 1 at time is associated with the command.
- ADDRESS - Pointer to HEX block in which the command is to be carried out.
- TIME - Time the command is to be performed.
- ACTION - Command or action code.
- h) FORMATION BLOCK
- PFORM - Pointer to first WINGMAN block in the list.

NUMFLTS - Number of WINGMAN blocks in the list.
Corresponds to number of flights in
the formation.

i) WINGMAN BLOCK

PNEXT - Pointer to next WINGMAN block in the
list.

PSB - Pointer to the SB block of the
flight.

j) ARCFTSTATUS BLOCK

PFLTDB - Pointer to FLTDB block

PMUNITIONS - Pointer to NUM block

PSTRTHX - Pointer to HEX block in which
current move begins

PENDHX - Pointer to HEX block in which
current move ends

PNXTHX - Pointer to next HEX block

PAIRBASE - Pointer to SB block of flight's home
air base.

CNTRLMODE -

PAIRTGT - Pointer to SB block of airborne target.

PGNDTGT - Pointer to SB block of ground
target.

NVMAIRCRAFT - Number of aircraft in the flight.

FLITELEG

INTERCEPTSTATUS

ALTUDCHNG

PROFILENDX

LANDING

ORBITSTAT

AIRCOMBAT

JAMSTAT

FUEL - Current fuel level. Real variable

ALTITUDE - Altitude in meters. Real variable
 SPEED - Speed. Real variable
 DIRECTION - Direction. Real variable
 k) MUM BLOCK
 PAG - Pointer to ground attack LOAD block
 NUMAG - Number of LOAD blocks in ground
 attack munitions list
 PAA - Pointer to air attack LOAD block
 NUMAA - Number of LOAD blocks in air attack
 munitions list
 l) LOAD BLOCK
 PNEXT - Pointer to next LOAD block in the
 list
 TYPE - Munitions class
 AMOUNT - Amount of munitions in tons
 PORDDB - Pointer to PAYLOAD block
 m) FLTDB BLOCK
 PBXFLDB - Pointer to next FLTDB block
 NRFLITE - Unique flight specification number
 PTYPLDS - Pointer to PAYLOAD data block
 Payload class 6002
 NOPYLDS - Number of PAYLOAD data blocks
 PTYAQDB - Pointer to ACQUISITION data block
 (AQDB), Class 6007)
 PTACDB - Pointer to aircraft specification
 data block (ACDB, CLASS 6003)
 MAXNOAC - Maximum number of aircraft in flight
 MINNOAC - Minimum number of aircraft in flight
 MULTAC - Multiples of aircraft required for
 flight
 PROFILE - Pointer to profile specification
 data block (Profiled blok, class 6005)

- SPFLTC(SPACE) Flight crusing speed in meters/second
(real)
- DISTSEP(SPACE) Flight separation distance in meters
(real)
- n) PAYLOAD BLOCK
- PNXTYPD - Pointer to next payload block
- NRPDCLS - Payload type, must be 3 or 4
3 = air-to-ground
4 = air-to-air
- MAXAMT - Maximum number of loads of this
payload
- MINAMT - minimum number of loads of this
payload
- MAXFIRERANGE Future use by an enhancement for
maximum fire range for engagements
greater than one hex
- PAYLDOB - Pointer to payload ID data block
(PAYLDOBLOK)
- o) PAYLDOBLOK BLOCK
- NEXT - Pointer to next ID block
- TYPEINDEX - Payload ID, unique within each
payload type
- p) AQDB BLOCK
- NEXT - Pointer to next AQDB block
- NRAQTYP - Unit type
- RANGE(SPACE) Acquisition range in meters (real)
- NOUSE1 - Not used
- NOUSE2 - Not used
- q) PROFILEDBLOK BLOCK
- PNXPRDB - Pointer to next PROFILEDBLOK
- NRPROFL - Profile identification number,
must be unique within the 6005 class

ALTCREN	-	Altitude of first leg in meters (real)
ALTOTGT	-	Altitude of second leg in meters (real)
ALTOAB	-	Altitude of third leg in meters (real)
r) <u>ACDB BLOCK</u>		
NEXT	-	Pointer to next ACDB block
NRATYPE	-	Aircraft type number
MAXSPEED	-	Maximum speed in meters/second (real)
CRUISESPEED		Cruising speed in meters/second (real)
MAXALTITUDE		Maximum altitude in meters (real)
MINALTITUDE		Minimum altitude in meters (real)
MAXCLIMBDIVE		Maximum climb/dive rate in meters/seconds (real)
FUELCONSUME		Fuel consumption rate in hexes/second (real)
ACQRANGE	-	Acquisition range in meters (real)
RADARCS	-	Radar cross sections in hexes (real)
ATTACKRADIUS		Attack radius in meters (real)
MAXFUEL	-	Maximum fuel load in hexes (real)

e. Linkages to Other Data Structures

f. Notes

G. BLUE STRUCTURES

In contrast to Red Structures which are dominated by threat planning related structures, Blue Structures are all related to some type of combat entity or player. Seven of those player types are connected to the BLUE C² tree, while the eight is connected to the passive target list.

1. ALLIED TACTICAL AIR FORCE (ATAF)

a. DATA BLOCK INDEX

C2

SB

SND

b. DESCRIPTION

The ALLIED TACTICAL AIR FORCE structure resides at the top of the blue C2 TREE (see E.2.f). It is currently used only to maintain the consistency of the blue command/control structure and does not initiate actions in the course of the simulation.

c. STRUCTURE OVERVIEW

1) STRUCTURE DIAGRAM (Figure III-21)

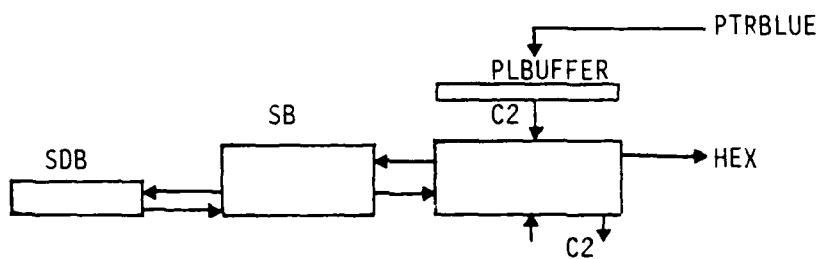


Figure III-21. Allied Tactical Air Force Structure Diagram

2) BLOCK DEFINITIONS

<u>PLBUFFER</u>	-	<u>C2 TREE BUFFER BLOCK</u> . Contains pointer to the tree and the side of the tree
<u>C2</u>	-	<u>COMMAND/CONTROL BLOCK</u> . Contains tree pointers, unit number, pointer to scoreboard (SB), unit type code and side
<u>SB</u>	-	<u>SCOREBOARD BLOCK</u> . Contains pointers to <u>C2</u> and <u>HEX</u> blocks, and the status display board (SDB). Also contains pointers

to acquisition devices, various unit status blocks and the future event list. Use varied with unit type.

SDB -

STATUS DISPLAY BOARD BLOCK. Contains subordinate, acquisition and order pointers. Use varies with unit type

d. BLOCK SPECIFICATIONS

1) PLBUFFER

PTRPL	VARIWORD
-------	----------

2) C2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITYPE	SIDE

NOTE: + ALTERNATE
DEFINITION OF FIELD

3) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATABASE	PABSTATUS
+	PARCFTSTAT
+	PBOCSTAT
+	PBTRYSTAT
+	STATUS

4) SDB

PSB	PSEEBUF
+	PSEE
SUBORDINATE	ORD
+	PRAID

e. FIELD DEFINITIONS

1) PLBUFFER BLOCK

PTRPL - Pointer to C2 Block
VARWORD - Side indicator for tree
1 = BLUE (NATO)
2 = RED (PACT)

2) C2 BLOCK

- Number of the unit. If negative, the unit is a passive target
PUP - Pointer to the C2 block of the unit's commander
PDOWN - Pointer to the C2 block of the unit's sibling
PSB - Pointer to the SB block of the unit
UNITTYPE - The unit's type code (see subsection 6).
SIDE - Unit affiliation
1 = BLUE (NATO)
2 = RED (PACT)

3) SB BLOCK

ADDRESS - Pointer to HEX block of the hex in which the unit is located
PC2 - Pointer to C2 block of the unit
PSDB - Pointer to the SDB block of the unit
PFEL - Pointer to future event list EVENT block
PACQ - Pointer to ACQBUF block. Used by CRC's for acquisition devices

ID	-	Identification number
DATABASE	-	If BOC or BTRY points to <u>ADSITEDB</u> block
PABSTATUS	-	Points to <u>ABSTATUS</u> block if the unit is an airbase
+PARCFTSTAT	-	Points to <u>ARCFTSTATUS</u> block if the unit is a flight or aircraft.
+PBOCSTAT	-	Points to <u>BOCSTAT</u> block if the unit is a battalion operations center
TPBTRYSTAT	-	Points to <u>BTRYSTAT</u> block if the unit is an anti-aircraft battery
+STATUS	-	alternative field definition

4) SDB BLOCK

PSB	-	Pointer to the unit's <u>SB</u> block
PSEEBUF	-	Pointer to <u>SEEBUF</u> block which is used by aircraft flight units to record targets seen and their perceived damage levels. This field definition is used only by flights
+PSEE	-	Pointer to <u>CRCEES</u> block which is used by <u>CRC</u> units to record the blue and red flights it sees. Also used by the RTC to point to its' assigned target list.

TARGET LISTBOOK

SUBORDINATE - Points to different types of subordinate description or target description blocks depending upon the unit type. Possible unit type vs field use combinations are as follows:

<u>UNIT TYPE</u>	<u>SUBORDINATE POINTS TO</u>
CRC	SUB
BOC	SVBUST
BTRY	FIREUNIT

ORD - Points to ORDERS block if the unit is a flight

+PRAID - Points to the RAIDBLOK if the unit is the Red theater commander

2. SECTOR OPERATIONS CENTERS (SOC)

a. DATA BLOCK INDEX

C2

SB

SDB

b. DESCRIPTION

The SECTOR OPERATIONS CENTER structures reside on the Blue C2 tree on the Blue C2 tree on the level below the ATAF. Like the ATAF, they are currently used only to maintain the consistency of the Blue command/control structure. SOC's do not initiate actions in the course of the simulation.

c. STRUCTURE OVERVIEW

1) STRUCTURE DIAGRAM (Figure III-22)

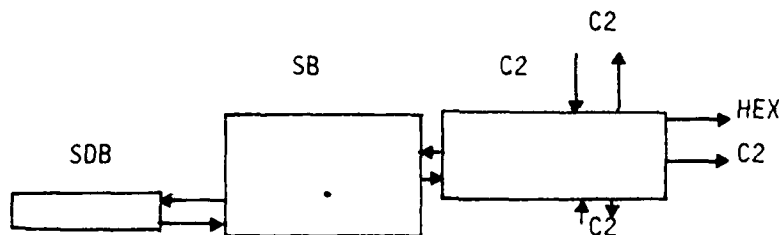


Figure III-22. Sector Operations Center Structure Diagram

2) BLOCK DEFINITIONS

- | | | |
|----------|---|---|
| PLBUFFER | - | <u>C2 Tree Buffer Block</u> . Contains pointer to the tree and to the side of the tree |
| C2 | - | <u>Command/Control Block</u> . Contains tree Pointers, unit number, pointer to scoreboard (<u>SB</u>), unit type code and side |
| SB | - | <u>Scoreboard Block</u> . Contains pointers to <u>C2</u> and <u>HEX</u> blocks, and the status display board (<u>SDB</u>). Also contains pointers to acquisition devices, various unit status blocks and the future EVENT list. Use varies with |

SDR - unit type
Status Display Board Block. Contains subordinate, acquisition and order pointers. Use varies with unit type

d. BLOCK SPECIFICATIONS

1) BLOCK DIAGRAMS

a) C2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITYPE	SIDE

NOTE: + ALTERNATE DEFINITION OF FIELD

b) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATABASE	PABSTATUS
+	PARCFTSTAT
+	PBOCSTAT
+	PBTRYSTAT
+	STATUS

c) SDB

PSB	PSEEBUF
+	PSEE
SUBORDINATE	ORD
+	PRAID

2) FIELD DEFINITIONS

a) C2 BLOCK

UNITNUMBER	-	Number of the unit. If negative, the unit is a passive target
PVP	-	Pointer to the <u>C2</u> block of the unit's commander
PDOWN	-	Pointer to the <u>C2</u> block of the unit's subordinate
PNEXT	-	Pointer to the <u>C2</u> block of the unit's sibling
PSB	-	Pointer to the <u>SB</u> block of the unit
UNITYPE	-	The unit's type code (see subsection 5))
SIDE	-	Unit affiliation 1 = BLUE (NATO) 2 = RED (PACT)

b) SB BLOCK

ADDRESS	-	Pointer to <u>HEX</u> block of the hex in which the unit is located
PC2	-	Pointer to <u>C2</u> block of the unit
PSDB	-	Pointer to the <u>SDB</u> block of the unit
PFEL	-	Pointer to future event list <u>EVENT</u> block
PACQ	-	Pointer to <u>ACQBUF</u> block. Used by <u>SRC's</u> for acquisition devices.
ID	-	Identification number

DATABASE	-	If BOC or BTRY points to <u>ADSITEDB</u> block
PABSTATUS	-	Points to <u>ABSTATUS</u> block if the unit is an airbase
+PARCFTSTAT	-	Points to <u>ARCFTSTATUS</u> block if the unit is a flight of aircraft
+PBOCSTAT	-	Points to <u>BOCSTAT</u> block if the unit is a battalion operation center
+PBTRYSTAT	-	Points to <u>BTRYSTAT</u> block if the unit is an anti-aircraft battery
+STATUS	-	Alternative field definition
c) <u>SDB BLOCK</u>		
PSB	-	Pointer to the unit's <u>SB</u> block
PSEEBUF	-	Pointer to <u>SEEBUF</u> block which is used by aircraft flight units to record targets seen and their perceived damage levels. This field definition is used only by flights.
+PSEE	-	Pointer to <u>CRCEES</u> block which is used by <u>CRC</u> units to record the blue and red flights it sees. Also used by the RTC to point to its assigned target list.
		<u>TARGET LIST BLOCK</u>

SUBORDINATE - Points to different types of subordinate description or target description blocks depending upon the unit type. Possible unit type vs vield use combinations are as follows:

<u>UNIT TYPE</u>	<u>SUBORDINATE POINTS TO</u>
CRC	SUB
BOC	SUBLIST
BTRY	FIREUNIT

ORD - Points to ODERS block if the unit is a flight

+ PRAID - Points to the RAIDBLOK if the unit is the Red theater commander

e. LINEAGES TO OTHER DATA STRUCTURES

Pointer to HEX block in which unit is located

f. NOTES

3. COMBAT REPORTING CENTERS (CRC)

a. DATA BLOCK INDEX

ACQBUF
ACQDEVICE
C2
CACSEEBLUE
CRCSEERED
CRCSEES
CRCSUBORD
SB
SDB
SEER
SUB
SUBTYPE

b. Description

The COMBAT REPORTING CENTERS are the highest ranking active players on the Blue side. In addition to the basic command control structures (C², SB and SDB), CRC's possess three unique lists - the PERCEPTIONS LIST, the SUBORDINATE LIST, and the ACQUISITION LIST.

The ACQUISITION LIST is used to keep track of acquisition devices belonging to the CRC. The SUBORDINATE LIST is used to keep track of the CRC's subordinate blue units (including FLIGHTS). These lists are stratified by device and subordinate type respectively.

The PERCEPTION LIST consists of two branches. One keeps track of Red Units and the Blue Units perceiving them. The other keeps track of Blue Units perceived by the CRC directly. It is important to note that the CRC may perceive enemy units through its subordinates. The overall configuration of the CRC structure is shown in the structure diagram.

c. Structure Overview

1) Structure Diagram (Figure III-23)

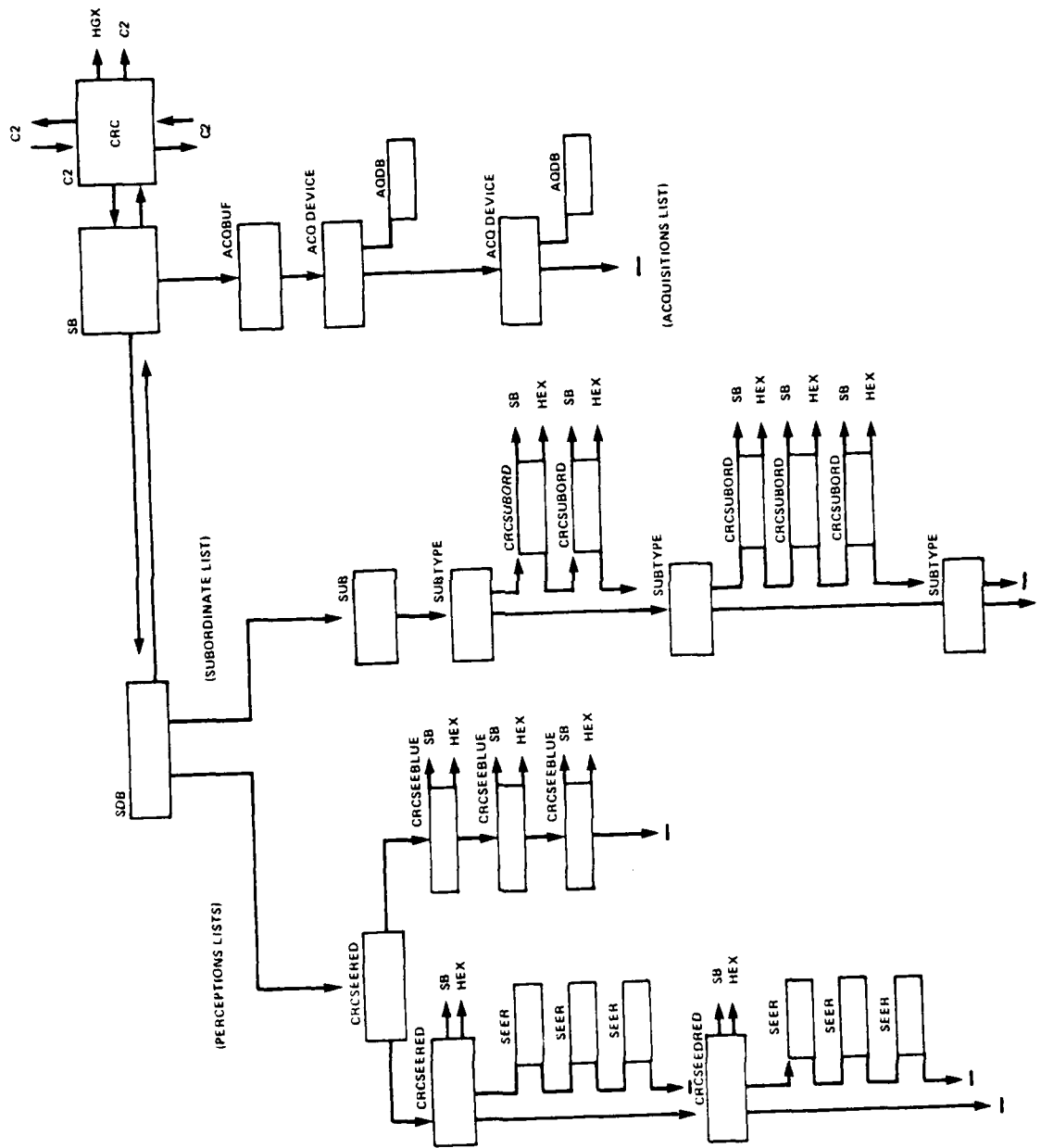


Figure III-23. Combat Reporting Center Structure Diagram

4368/79W

2) Block Definitions

<u>C²</u>	-	<u>COMMAND/CONTROL BLOCK.</u> Contains tree pointers, unit number, pointer to scoreboard (SB), unit type code and side.
<u>SB</u>	-	<u>SCOREBOARD BLOCK.</u> Contains pointers to C ² and HEX blocks, and the status display board (SDB). Also contains pointers to acquisition devices various unit status blocks and the future event list. Use varies with unit type.
<u>SDB</u>	-	<u>STATUS DISPLAY BOARD BLOCK.</u> Contains subordinate, acquisition and order pointers. Use varies with unit type.
<u>ACQBUF</u>	-	<u>ACQUISITION DEVICE LIST BUFFER.</u> Contains a pointer to the device list and the number of devices (ACQDEVICE BLOCKS) in the list.
<u>ACQDEVICE</u>	-	<u>ACQUISITION DEVICE DESCRIPTION BLOCK.</u> Contains device code, operating status and a pointer to the next ACQDEVICE block in the list.
<u>AQDB</u>	-	<u>ACQUISITION DEVICE DATA BASE BLOCK.</u> Basic data on specific acquisition device.
<u>SUB</u>	-	<u>SUBORDINATE LIST BUFFER.</u> Contains a pointer to the subordinate list and the number of subordinate types (SUBTYPE BLOCKS) in the list.

<u>SUBTYPE</u>	-	<u>SUBORDINATE TYPE BLOCK.</u> Contains a subordinate type code, the number of subordinates of that type, a pointer to the next SUBTYPE and a pointer to a list of specific subordinate unit blocks (CRCSUBORD).
<u>CRCSUBORD</u>	-	<u>CRC SUBORDINATE BLOCK.</u> Contains unit ID of the subordinate, a pointer to its SB block and HEX block, and a pointer to the next CRCSUBORD block in the list.
<u>CRCSEES</u>	-	<u>CRC PERCEPTIONS LIST BUFFER BLOCK.</u> Contains pointers to Red and Blue branches and the number of units (CRCSEERED and CRCSEEBLUE BLOCKS) in each branch.
<u>CRCSEERED</u>	-	<u>RED UNIT PERCEIVED BLOCK.</u> Contains a description of Red Unit perceived. Includes pointers to its' SB and HEX blocks as well as its' direction of movement. Also contains a pointer to a list of Blue units which can see it.
<u>SEER</u>	-	<u>BLUE PERCEPTION UNIT BLOCK.</u> Contains unit ID of the Blue Unit perceiving a Red Unit.
<u>CRCSEEBLUE</u>	-	<u>BLUE UNIT PERCEIVED BLOCK.</u> Contains the unit ID, SB and HEX block pointers of a Blue Unit perceived by the CRC.

d. Block Specifications

1) Block Diagrams

a) C²

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

Note: + Alternate Definition of field.

b) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATABASE	PABSTATUS
+	PARCFTSTAT
+	PBOCSTAT
+	PBTRYSTAT
+	STATUS

c) SDB

PSB	PSEEBUF
+	PSEE
SUBORDINATE	ORD
+	PRAID

d) ACQBUF

PTRACQ	NUMDEV
--------	--------

e) ACQDEVICE

PNEXT	TYPE
WORKING	PACQDB
JAM	LEVEL

f) AQDB

NEXT	NRAQTYP
RANGE(SPACE)	
NOUSE1	
NOUSE2	

g) SUB

PSUB	NUMBER
------	--------

h) SUBTYPE

PNEXT	TYPE
PTRSUB	NUMBER

i) CRCSUBORD

PNEXT	ID
PSB	ADDRESS
WORD 1	WORD 2

j) CRCSEES

REDSEE	NUMBERD
BLUESEE	NUMBLUE

k) CRCSEERED

PNEXT	ID
PSB	ADDRESS
RPT	HUNTER
DIRECTION (SPACE)	
PNX	NUMSEE

l) SEER

PNEXT	ID
-------	----

m) CRCSEEBLUE

PNEXT	ID
PSB	ADDRESS

2) FIELD DEFINITIONS

a) C² BLOCK

- | | | |
|------------|---|--|
| UNITNUMBER | - | Number of the unit. If negative the unit is a passing target. |
| PVP | - | Pointer to the <u>C²</u> block of the unit's commander. |
| PDOWN | - | Pointer to the <u>C²</u> block of the unit's subordinate. |

PNEXT	-	Pointer to the <u>C²</u> block' of the unit's commander.
PSB	-	Pointer to the SB block of the unit
UNITTYPE	-	The unit's type code (see subsection f)
SIDE	-	Unit affiliation. 1 = Blue (NATO) 2 = Red (PACT)

b) SB BLOCK

ADDRESS	-	Pointer to HEX block of the HEX in which the unit is located.
PC2	-	Pointer to C ² block of the unit
PSDB	-	Pointer to the SDB block of the unit
PFEL	-	Pointer to future event list EVENT BLOCK.
PACQ	-	Pointer to ACQBUF BLOCK. Used by CRC's for acquisition devices.
ID	-	Identification number
DATABASE	-	If BOC or BTRY points to ADSITEDB block
PABSTATUS	-	Points to ABSTATUS block if the unit is an airbase
+PARCFTSTAT-		Points to ARCFTSTATUS block if the unit is a flight of aircraft.
+PBOCSTAT	-	Points to BOCSTAT block if the unit is a battalion operations center
+PBTRYSTAT	-	Points to BTRYSTAT block if the unit is an anti-aircraft battery

+STATUS - Alternative field definition
 +PBTRYSTAT - Points to BOCSTAT block if the unit is an anti-aircraft
 +STATUS - Alternative field definition.

c) SDB BLOCK

PSB - Pointer to the unit's SB block.
 PSEEBUF - Pointer to SEEBUF block which is used by aircraft flight units to record targets seen and their perceived damage levels. This field definition is used only by flights.

+PSEE - Pointer to CRCEES block which is used by CRC units to record the Blue and Red flights it sees. Also used by RTC to point to its' assigned targets list

SUBORDINATE - Points to different types of subordinate description or target description blocks depending upon the unit type. Possible unit type vs. field use combinations are as follows:

<u>UNIT TYPE</u>	<u>SUBORDINATE POINTS TO</u>
CRC	SUB
BOC	SUBLIST
BTRY	FIREUNIT

ORD - Points to ORDERS block if the unit is a flight

+PRAID - Points to RAIDBLOK if the unit is the Red Theater Commander.

- d) ACQBUF BLOCK
- PTRACQ - Pointer to ACQDEVICE block.
- NUMDEV - Number of ACQDEVICE blocks in the list.
- e) ACQDEVICE BLOCK
- PNEXT - Pointer to the next ACQDEVICE block in the list.
- TYPE - Acquisition device type code
- WORKING -
- PACQDB - Pointer to ACDB block.
- JAM -
- LEVEL -
- f) AQDB BLOCK
- NEXT - Pointer to next AQDB block in data base
- NRAQTYP - Acquisition device type code
- RANGE - Range of devide. Real variable
- NOUSE1 -
- NOUSE2 -
- g) SUB BLOCK
- PSUB - Pointer to SUBTYPE block
- NUMBER - Number of SUBTYPE blocks in the list
- h) SUBTYPE BLOCK
- PNEXT - Pointer to next SUBTYPE block
- TYPE - Subordinate type code
- PTRSUB - Pointer to CRCSUBORD block
- NUMBER - Number of CRCSUBORD blocks in the list
- i) CRCSUBORD BLOCK
- PNEXT - Pointer to next CRCSUBORD block
- ID - Unit ID

	PSB	-	Pointer to Unit's SB block
	ADDRESS	-	Pointer to Unit's HEX block.
	WORD1	-	
	WORD2	-	
j)	<u>CRCEES BLOCK</u>		
	REDSEE	-	Pointer to CRCSEERED block
	NUMRED	-	Number of CRCSEERED blocks in the list.
	BLUESEE	-	Pointer to CRCSEEBLUE blocks
	NUMBLUE	-	Number of CRCSEEBLUE blocks in the list.
k)	<u>CRCSEERED BLOCK</u>		
	PNEXT	-	Pointer to next CRCSEERED block
	ID	-	Unit ID.
	PSB	-	Pointer to Red Unit's SB block
	ADDRESS	-	Pointer to Red Unit's HEX block
	RPT	-	
	HUNTER	-	
	DIRECTION	-	Direction of travel, real variable
	PNX	-	Pointer to SEER block
	NUMSEE	-	Number of SEER blocks in the list
l)	<u>SEER BLOCK</u>		
	PNEXT	-	Pointer to next SEER block
	ID	-	Unit ID of Blue perceiver
m)	<u>CRCSEEBLUE BLOCK</u>		
	PNEXT	-	Pointer to next CRCSEEBLUE block
	ID	-	Unit ID
	PSP	-	Pointer to Unit's SB block
	ADDRESS	-	Pointer to Unit's HEX block

e. LINKAGES TO OTHER DATA STRUCTURES

The CRC STRUCTURE is linked to both the HEX ADDRESS TREE and the C^2 TREES of both sides.

f. NOTES

4. BLUE AIRBASES (B-AB)

a. Data Block Index

ABINFO

ABQUEDB

ABSTATUS

ACDB

C²

QUEUES

QUESTAT

SB

b. Description

BLUE AIRBASE STRUCTURES are designed to keep track of the number and type of aircraft on the AIRBASE as well as the status of each aircraft type in terms launch capability. The overall configuration of these structure is shown in the structure diagram.

c. Structure Overview

1) Structure Diagram (Figure III-24)

2) Block Definitions

- | | | |
|----------------|---|---|
| C ² | - | COMMAND CONTROL BLOCK. Contains unit no., type and side, and C ² PTRS. |
| SB | - | SCOREBOARD BLOCK. Contains HEX address PTR, Unit ID, PTR to future event list for the unit, and ABSTATUS PTR. |
| SDB | - | STATUS DISPLAY BLOCK. Not used by airbase. |
| ABSTATUS | - | AIRBASE STATUS BLOCK. Contains PTRS to the airbase information list (ABINFO) and the QUEUES list for each aircraft type on base. |
| ABINFO | - | AIRCRAFT ON BASE INFORMATION BLOCK. Contains aircraft type, no on hand and no. in each service queue, and a PTR to A/C type's ACDB. |

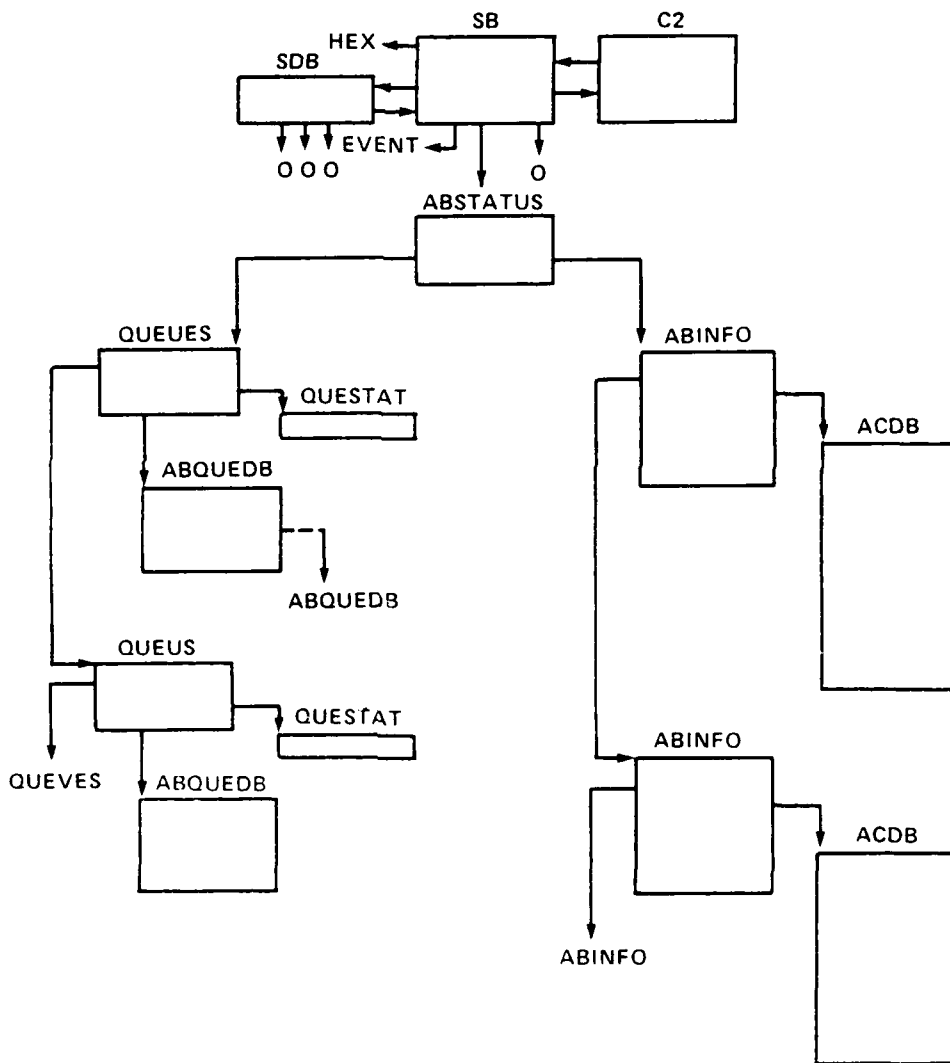


Figure III-24. Blue Airbase Structure Diagram

4368/79W

- QUEUES - AIRCRAFT SERVICE QUEUE BLOCK
- ABQUEDB - AIRBASE QUEUE DATA BASE BLOCK
- QUESTAT - QUEUE STATUS BLOCK.
- ACDB - AIRCRAFT DATA BASE BLOCK. Operational parameters of A/C type.

d. Block Specification

1) Block Diagrams

a) C²

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

b) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATABASE	PABSTATUS

c) ABSTATUS

PACTAB	NOACTAB
PTR2QUES	NOACONAB
ABDAMAGE (SPACE)	

d) QUEUES

NEXT	QUENUM
PTR	NUMBER
PQDB	PQUESTAT

e) QUESTAT

VALUE (SPACE)

f) ABQUEDB

PNEXT	CLASS
VALUE1 (SPACE)	
VALUE2 (SPACE)	
VALUE3 (APCE)	

g) ABINFO

NEXT	NRACTYP
NOACOH	PTRACDB
NORMRQ	
NORMEARMQ	
NOREFVELQ	
NOLAUNCHQ	

h) ACDB

NEXT	NRACTYPE
MAXSPEED (SPACE)	
CRUISESPEED (SPACE)	
MAXALTITUDE (SPACE)	
MINALTITUDE (SPACE)	
FUELCONSUME (SPACE)	
ACQRANGE (SPACE)	
RADARCS (SPACE)	
ATTACKRADIUS (SPACE)	
MAXFUEL (SPACE)	

2) Field Specifications

a) C² BLOCK

- UNITNUMBER - Unit number
- PUP - Pointer to C² block of unit's commander
- PDOWN - Pointer to C² block of unit's subordinate
- PSB - Pointer Unit's SB block
- PNEXT - Pointer to Unit's sibling. C² block
- UNITTYPE - Unit type code (220)
- SIDE - Unit affiliation
 - 1 = Blue (NATO)
 - 2 = Red (PACT)

- b) SB BLOCK
- | | | |
|----------|---|--|
| ADDRESS | - | Pointer to HEX block in which unit is located. |
| PC2 | - | Pointer to C ² block of the unit. |
| PSDB | - | Pointer to SDB block (Inactive) |
| PFEL | - | Pointer to future event list for the unit. |
| PACQ | - | Pointer to acquisition devices (inactive) |
| ID | - | Unit ID number |
| DATABASE | - | Pointer to database block (inactive) |
- c) ABSTATUS BLOCK
- | | | |
|----------|---|---|
| PACTAB | - | Pointer to ABINFO block. |
| NOACTAB | - | Number of ABINFO blocks in the list corresponds to number of aircraft on the base |
| PTR2QUES | - | Pointer to QUEUES BLOCK |
| NOACONAB | - | Number of QUEUES blocks in the list. Corresponds to number of aircraft types on the base. |
| ABDAMAGE | - | Damage level of base. Real variable. |
- d) QUEUES BLOCK
- | | | |
|---------|---|---|
| NEXT | - | Pointer to next QUEUES block in the list. |
| QUENUM | - | QUEUE number (2 = ready queue) |
| PTR | - | |
| NUMBER | - | |
| PQDB | - | Pointer to ABQUEDB block |
| QPESTAT | - | Pointer to QUESTAT block |

- e) QUESTAT BLOCK
 - VALUE - Unknown, real variable
- f) ABQUEDB BLOCK
 - PNEXT - Pointer to next ABQUEDB block
 - CLASS - Aircraft class
 - VALUE1 -
 - VALUE2 -
 - VALUE3 -
- g) ABINFO BLOCK
 - NEXT - Pointer to next ABINFO block
 - NRACTYP - Aircraft type code
 - NOACOH - Number of aircraft of type (NRACTYP) on hand on the base.
 - PTRACDB - Pointer to ACDB for the aircraft type (NRACTYP)
 - NORMRQ - Number in repair queue
 - NOREARMQ - Number in rearm queue
 - NOLAUNCHQ - Number in launch queue
- h) ACDB BLOCK
 - NEXT - Pointer to next ACDB block in data base (not used in this context)
 - NRACTYPE - Aircraft type code
 - MAXSPEED - Maximum speed. Real variable
 - CRUISESPEED - Cruising speed. Real variable
 - MAXALTITUDE - Maximum altitude. Real variable
 - MAXCLIMBDIVE - Maximum rate of altitude change. Real variable.
 - FUELCONSUME - Fuel consumption rate. Real variable.
 - ACQRANGE - Acquisition range. Real variable.

RADARCS - Radar cross-section. Real variable.
ATTACKRADIUS - Maximum attack range of aircraft. Real variable.
MAXFUEL - Maximum fuel capacity. Real variable.

e. Linkages to Other Data Structures

f. Notes

QUEUES blocks and their related lists are not used at present.

5. Blue Flights (B-FLT)

a. Data Block Index

ACDB
AQDB
ARCFSAW
ARCFTSTATUS
C²
COMMAND
FLTDB
FORMATION
LOAD
MUN
ORDERES
PAYDDBLOK
PAYLOAD
SB
SEEBUF
WINGMAN

b. Description

The BLUE FLIGHT DATA STRUCTURES control the actions of blue flights. In addition to the three command/control blocks C²; SB and SDB, BLUE FLIGHTS also use three lists. These lists include: the PERCEPTIONS LIST, the ORDERS LIST and the FLIGHT STATUS LIST.

The PERCEPTIONS LIST is composed of a buffer and a singly-linked list of ARCFSAW blocks. These ARCFSAW blocks contain information on Red flights perceived by the Blue FLIGHT. This information includes the location of the Red targets. When the Blue FLIGHT returns to its' airbase the perceived damage to Red Flights is transferred to the Blue CRC.

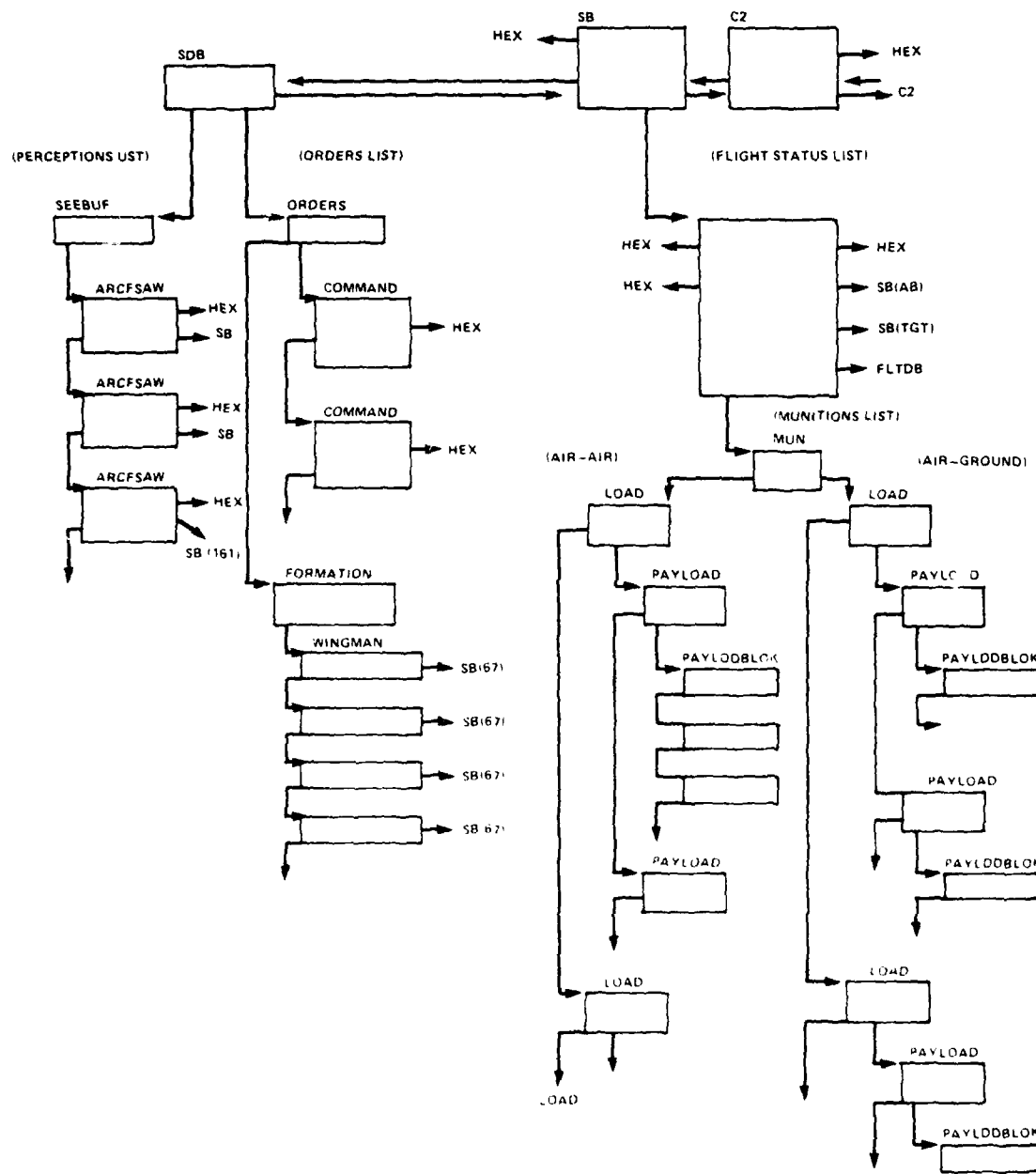
The ORDERS LIST is composed of two buffered lists. The first is made-up of up to six COMMAND blocks which specify the actions to be taken by the flight at various points in its' mission. These COMMAND blocks determine the flight geometry and mission profile for the flight. The second list is made up to WINGMAN blocks which contain pointers. To the SB blocks of other flights in the other Blue FLIGHTS in the formation.

The FLIGHT STATUS LIST consists of an ARCFTSTATUS block which tracks flight status and a set of two NUMITIONS LISTS which keep track of air-to-air and air-to-ground ordinance carried by the Red Flight. The ARCFSTATUS block points to an FLTOB block which is the core of a FLIGHT DATA BASE STRUCTURE. This structure is used as a template for construction of flights of specified types. It provides the basic aircraft characteristics and initial payload levels used to create and operate the flight.

Both the FLIGHT DATA STRUCTURES and the FLIGHT DATA BASE STRUCTURES are shown in the structure diagrams.

c. Structure Overview

1) Structure Diagrams (Figures III-25 & III-26)



4368/79W

Figure III-25. Blue Flight Structure Diagram

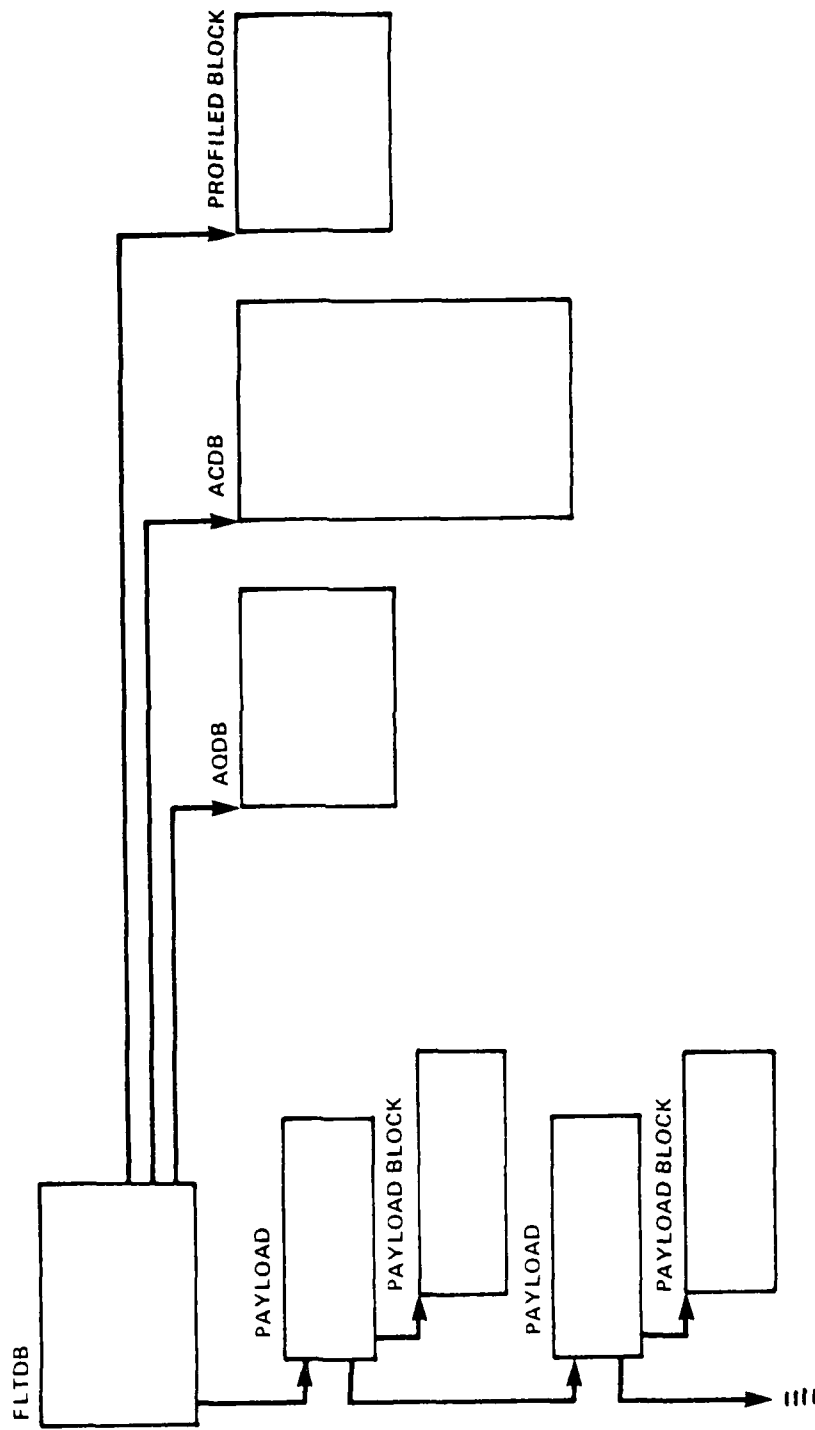


Figure III-26. Flight Data Base Structure Diagram

4368/79W

2) Block Definitions

COMMAND/CONTROL BLOCKS

<u>C²</u>	-	<u>COMMAND/CONTROL BLOCK.</u> Contains Unit No., type, side, and C ² PTRS.
<u>SB</u>	-	<u>SCOREBOARD BLOCK.</u> Contains HEX address, unit ID, PTR to FEL and STAT
<u>SDB</u>	-	<u>STATUS DISPLAY BLOCK.</u> Contains PTRS to perceptions list (SEEBUF) and orders list (ORDERS). Subordinate PTR not used by Red FLTS.

FLIGHT STATUS LIST

<u>ARCFTSTATUS</u>	-	<u>FLIGHT STATUS BLOCK.</u> Contains basic flight status information includes PTRS to startin, ending and next HEX address in current move. Also includes PTRS to homebase and TGT SB's a status word, and current fuel, altitude, speed, and direction.
--------------------	---	--

<u>MUN</u>	-	<u>MUNITIONS LIST BUFFER BLOCK.</u> Buffer for air to air and air to ground munitions lists.
------------	---	--

<u>PAYLOAD</u>	-	<u>PAYLOAD CLASS DESCRIPTION BLOCKS.</u> Basic payload parameters for class of ordnance. Contains class type, max and min amounts and max fire range. Also includes PTR to a list of attached ordnance types of the same class.
----------------	---	---

<u>PAYDDBLOK</u>	-	<u>PAYLOAD TYPE DATA BASE BLOCK.</u> Contains PTR to next PAYDDBLOK and type of ordinance.
<u>LOAD</u>	-	<u>PAYLOAD LIST BUFFER.</u> Used to break payloads into types and keep track of ammunition load weight.
<u>PERCEPTIONS LIST</u>		
<u>SEEBUF</u>	-	<u>PERCEPTIONS LIST BUFFER BLOCK</u>
<u>ARCFSAW</u>	-	<u>AIRCRAFT PERCEPTION BLOCK.</u> Contains information on entity perceived by a flight. Includes PTR to HEX and SB of entity.
<u>ORDERS LIST</u>		
<u>ORDERS</u>	-	<u>ORDERS LIST BUFFER BLOCK.</u> Con- tains no. of orders remaining.
<u>COMMAND</u>	-	<u>COMMAND DESCRIPTION BLOCK.</u> Describes command to be followed by flight at specified address. Up to six in the list.
<u>FORMATION</u>	-	<u>FORMATION BUFFER BLOCK.</u> Con- tains number of flts in the formation to which the flight belongs.
<u>WINGMAN</u>	-	<u>WINGMAN LIST BLOCK.</u> Contains PTR to other flts in the forma- tion.

FLIGHT DATA BASE

FLTDB -

FLIGHT DATA BASE BLOCK. Contains basic flight description including no. of payloads, maximum no. of A/C, minimum No. of A/C, and Multic SPFLTC(SPACE, DISTSER(SPACE)

PAYLOAD -

PAYLOAD DESCRIPTION BLOCK. Contains payload class (NRPDCLS) max and min amount of payload, and max fire range

PAYLDBLOK -

PAYLOAD TYPE BLOCK. Contains payload type index

AQDB -

ACQUISITION DEVICE DATA BASE BLOCK. Contains type index and range of acquisition device.

ACDB -

AIRCRAFT DATA BASE BLOCK. Contains A/C characteristics such as speed, max range etc.

PROFILEDBLOK -

MISSION PROFILE BLOCK. Contains flight altitude levels for three phases of mission - ALT to corridor entrance, ALT to TGT, and ALT from TGT to Airbase.

1) Block Diagrams

a) C^2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

b) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATABASE	PABSTATUS
	+ PARCFTSTAT
	+ PBOCSTAT
	+ PBTRYSTAT
	+ STATUS

c) SDB

PSB	PSEEBUF
	+ PSEE
SUBORDINATE	ORD
	+ PRAID

d) SEEBUF

PTRSEE	NUNITS
--------	--------

e) ARCFSAW

PNEXT	PSB
ADDRESS	TYPE
DAMAGE (SPACE)	

f) ORDERS

PTRFORMS	PTRACT
----------	--------

g) COMMAND

PNEXT	NUMACTS
TMFLG	ADDRESS
TIME (SPACE)	
ACTION	

h) FORMATION

PFORM	NUMFLTS
-------	---------

i) WINGMAN

PNEXT	PSB
-------	-----

AD-A107 916

BDM CORP MCLEAN VA

F/6 15/3

MODULAR AIR DEFENSE EFFECTIVENESS MODEL, PROGRAM DOCUMENTATION --ETC(U)

JAN 80 M FILTEAU, B MACALEER, J T HAWKINS

DNA001-79-C-0230

UNCLASSIFIED

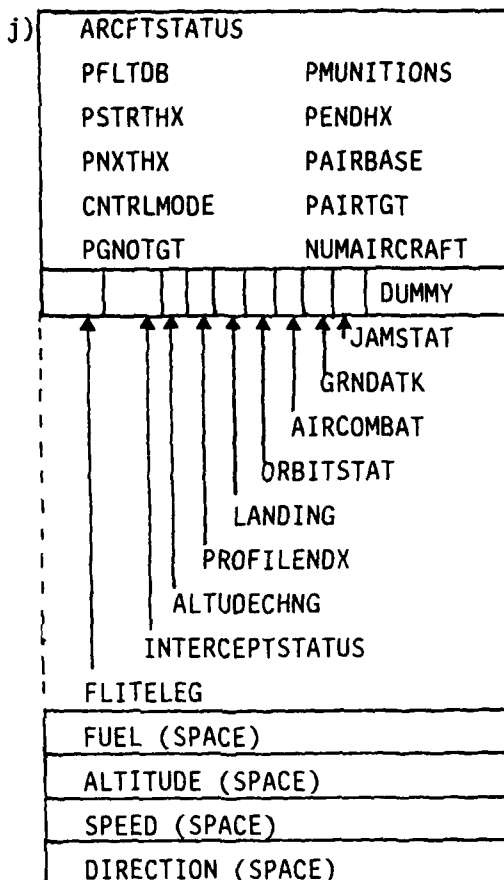
BDM/W-79-646-TR-VOL-2

DNA-5594F-2

NL

3 of 6
410794F





k) MUN

PAG	NUMAG
PAA	NUMAA

l) LOAD

PNEXT	TYPE
AMOUNT	PORDOB

1) FLTDB

PNXFLDB	NRFLITE
PTYPLDS	NOPYLDS
PTYAQDB	PTACDB
MAXNOAC	MINNOAC
MULTAC	PROFILE
SPFLTC (SPACE)	
DISTSEP (SPACE)	

n) PAYLOAD

PNXTYPD	NRPDCLS
MAXAMT	MINAMT
MAXFIRERANGE	PAYLODB

o) PAYLODBLOK

NEXT	TYPEINDEX
------	-----------

p) AQDB

NEXT	NRAQTYP
RANGE (SPACE)	
NOUSE1	
NOUSE2	

q) PROFILEDBLOK

PNXPRDB	NRPROFL
ALTCREN (SPACE)	
ALTOTGT (SPACE)	
ALTOAB (SPACE)	

r) ACDB

NEXT	NRACTYPE
MAXSPEED (SPACE)	
CRUISESPEED (SPACE)	
MAXALTITUDE (SPACE)	
MINALTITUDE (SPACE)	
MAXCLIMBDIVE (SPACE)	
FUELCONSUME (SPACE)	
ACQRANGE (SPACE)	
RADARCS (SPACE)	
ATTACKRADIUS (SPACE)	
MAXFUEL (SPACE)	

2) Field Definitions

a) C² BLOCK

UNITNUMBER - Number of the unit.
 PUP - Pointer to the C² block of the unit's commander.

PDOWN - Pointer to the C² block of the unit's sibling.
 PSB - Pointer to the SB block of the unit.
 UNITTYPE - The unit's type code (128)
 SIDE - Unit affiliation
 2 = Red (PACT)

b) SB BLOCK

ADDRESS - Pointer to HEX block of the HEX in which the unit is located
 PL2 - Pointer to C² block of the unit.
 PSDB - Pointer to the SDB block of the unit.
 PFEL - Pointer to future event list EVENT block.
 PACQ - Pointer to ACQBUF block. Used by CRC's for acquisition devices.
 ID - Identification number.
 DATABASE - If BOC or BTRY points to ADSITEDB block
 PABSTATUS - Points to ABSTATUS block if the unit is an airbase
 +PARCFTSTAT - Points to ARCFTSTATUS block if the unit is a flight of aircraft
 +PBOCSTAT - Points to BOCSTAT block if the unit is a battalion operations center.
 +PBTRYSTAT - Points to BTRSTAT block if the unit is an anti-aircraft battery
 +STATUS - Alternative field definition.

c) SDB BLOCK

PSB - Pointer to the unit's SB block
 PSEEBUF - Pointer to SEEBUF block which is used by aircraft flight units to record targets seen and their perceived damage levels. This field definition is used only by flights.

+PSEE - Pointer to CRCEES block which is used by CRC units to record the blue and red flights it sees. Also by the RTC as a pointer to it's assigned target list.

SUBORDINATE Points to different types of subordinate description or target description blocks depending upon the unit type. Possible unit type vs. field use combination are as follows:

<u>UNIT TYPE</u>	<u>SUBORDINATE POINTS TO</u>
CRC	SUB
BOC	SUBLIST
BTRY	FIREUNIT

ORD - Points to ORDERS block if the unit is a flight.

+PRAID - Points to the RAIDBLOK if the unit is the Red Theater Commander.

d) SEEBUF BLOCK

PTRSEE - Pointer to ARCFSAW block (first in list)

	NUNITS	-	Number of ARCFSAW blocks in the list. Corresponds to number of targets perceived.
e)	<u>ARCFSAW BLOCK</u>		
	PNEXT	-	Pointer to next ARCFSAW block in the list.
	PSB	-	Pointer to SB block of target.
	ADDRESS	-	Pointer to HEX block in which target is located.
	TYPE	-	Unit type code of target
	DAMAGE	-	Perceived damage level of the target. Real variable.
f)	<u>ORDERS BLOCK</u>		
	PTRFORMS	-	Pointer to FORMATION block
	PTRACT	-	Pointer to COMMAND block.
	PNEXT	-	Pointer to next command block.
	NUMACTS	-	Number of the command in the list.
	TMFLG	-	Time flag. If 1 at time is associated with the command.
	ADDRESS	-	Pointer to HEX block in which the command is to be carried out.
	TIME	-	Time the command is to be performed.
	ACTION	-	Command or action code
h)	<u>FORMATION BLOCK</u>		
	PFORM	-	Pointer to first WINGMAN block in the list.
	NUMFLTS	-	Number of WINGMAN blocks in the list. Corresponds to number of flights in the formation.

i) WINGMAN BLOCK

PNEXT - Pointer to next WINGMAN block
in the list

PSB - Pointer to the SB block of
the flight.

j) ARCFTSTATUS BLOCK

PFLTDB - Pointer to FLTDB block

PMUNITIONS - Pointer to MUN block

PSTRTHX - Pointer to HEX block in which
current move begins

PENDHX - Pointer to HEX block in which
current move ends

PNXTHX - Pointer to next HEX block

PAIRBASE - Pointer to SB block of flight's
home airbase.

CNTRLMODE -

PAIRTGT - Pointer to SB block of airborne
target.

PGNDTGT - Pointer to SB block of ground
target

NUMAIRCRAFT - Number of aircraft in the flight

FLITELEG

INTERCEPTSTATUS

ALTUDECHNG

PROFILENDX

LANDING

ORBITSTAT

AIRCOMBAT

BRNDATK

FUEL - Current fuel level. Real vari-
able

	ALTITUDE	-	Altitude in meters. Real variable
	SPEED	-	Speed. Real variable
	DIRECTION	-	Direction. Real variable.
k)	<u>MUN BLOCK</u>		
	PAG	-	Pointer to ground attack LOAD block
	NUMAG	-	Number of LOAD blocks in ground attack munitions list.
	PAA	-	Pointer to air attack LOAD block
	NUMDA	-	Number of LOAD blocks in air attack munitions list.
l)	<u>LOAD BLOCK</u>		
	PNEXT	-	Pointer to next LOAD block in the list
	TYPE	-	Munitions class
	AMOUNT	-	Amount of munitions in tons
	PORDDB	-	Pointer to PAYLOAD block
m)	<u>FLTDB BLOCK</u>		
	PNXFLDB	-	Pointer to next FLTDB block
	NRFLITE	-	Unique flight specification number
	PTYPLDS	-	Pointer to payload data block payload class 6002
	NOPYLDS	-	Number of payload data blocks
	PTYAQDB	-	Pointer to acquisition data block (AQDB, Class 6007)
	PTACDB	-	Pointer to aircraft specification data block (ACDB, Class 6003)
	MAXNOAC	-	Maximum number of aircraft in flight

	MINNOAC	-	Minimum number of aircraft in flight
	MULTAC	-	Multiples of aircraft required for flight
	PROFILE	-	Pointer to profile specification data block (PROFILEDBLOK), Class 6005)
	SPFLT(CSPACE)		Flight cruising speed in meters/second (real)
	DISTSEP(CSPACE)		Flight separation distance in meters (real)
n)	<u>PAYLOAD BLOCK</u>		
	PNXTYPD	-	Pointer to next PAYLOAD block
	NRPDCLS	-	Payload type, must be 3 or 4 3 = air to ground 4 = air to air
	MAXAMT	-	Maximum number of loads of this payload
	MINAMT	-	Minimum number of loads of this payload
	MAXFIRERANGE		Future use by an enhancement for maximum fire range for engagements greater than one hex
	PAYLDDB	-	Pointer to PAYLOAD ID data block (PAYLDDBLOK)
o)	<u>PAYLDDBLOK BLOCK</u>		
	NEXT	-	Pointer to next ID block
	TYPEINDEX	-	Payload ID, unique within each payload type
p)	<u>AQOB BLOCK</u>		
	NEXT	-	Pointer to next AQOB block
	NRAQTYP	-	Unit type

6. BATTALION OPERATIONS CENTERS (BOC)

a. Data Block Index

ABSITEDB

BOCSTAT

C2

DIB

DIL

PAL

PERLIST

SB

SDB

SOURCE

SUBLIST

b. Description

The air defense information for the Battalion Operations Center (BOC) in MADEM is stored in the Digested Information List (DIL). The DIL is integrated with the subordinates of the BOC to create the Threat-Defense Analysis Structure for each BOC.

The DIL represents the perceived threat to the BOC and is pointed to from the BOC's Status Board (BOCSTAT). The list of subordinates to the BOC are pointed to from the BOC Status Display Board (SDB). The BOC links its available subordinates to the targets producing the Threat-Defense Analysis Structure, made up of Possibilities and Allocations Lists (PAL) blocks. This structure of PAL blocks is in the form of a sparse matrix. Conceptually, the rows of the matrix shows what targets (DIL's) are associated with each subordinate (SUBLIST) so the subordinate can be maximally allocated. The columns are each target, DIL, and the subordinates that are currently associated with it. The elaborate linking allows easy addition or deletion from the matrix as well as ease of searching for the information needed.

The DIL's are divided up into three different configurations: (1) Active DIL list, (2) Passive DIL list, (3) Force out queue. This allows

the threats, DIL's, to be handled in a systematic way. All the DIL configurations are pointed to from the BOCSTATUS.

The Active DIL (ADIL) is a doubly linked circular list of all the threats actively being considered by the BOC. The DIL's in the ADIL are also doubly threaded creating a list in sequence of priorities. Each DIL in the ADIL is considered one at a time in the list in increments controlled by the Scheduled Digest event. The Air Defense Data Base controls the maximum number of DIL's a BOC can have in its ADIL.

When a BOC fully assigns the responsibility of a threat, DIL, its subordinates, the DIL is taken out of the ADIL and put into the Passive DIL list (PDIL). The PDIL is a doubly linked list of DIL blocks with a pointer to the head of the list.

If a DIL in the ADIL is forced out because the need to bring in a DIL of a higher priority, it is put in the Force Out Queue (FOQ). The FOQ is a double linked list with a head and tail pointer to the list. This is an overflow list and the DIL will eventually be brought back into the ADIL when there is an opening. The head and tail pointers allow access from either end of the list to get the best DIL out of the FOQ.

Each DIL has an associated Digested Information Block (DIB) with it. This holds the current information that the BOC has on the target. When the DIL is in the Active OIL List a new DIB is created and compared to the old one each time the DIL (target) is examined. The DIL's are also associated with the Perceptions List, a PERLIST block from the Perceptions List and a DIL are linked up. The information in the PERLIST is from the Perception event and is used with the DIL to represent the information on a target.

c. Structure Overview

1) Structure Diagrams (Figure III-27, III-28, III-29)

2) Block Definitions

C²

-

COMMAND/CONTROL BLOCK. Contains unit no., type, side and C² PTRS.

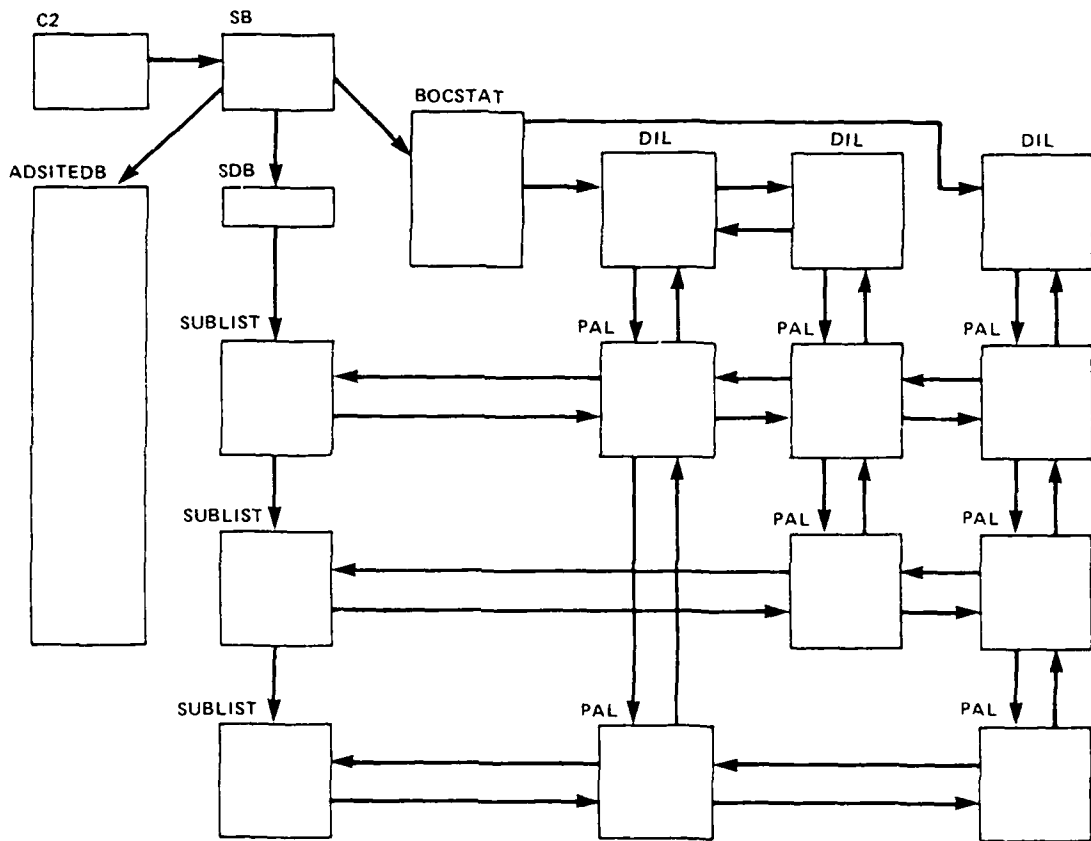


Figure III-27. Battalion Operations Center Structure Diagram

4368/79W

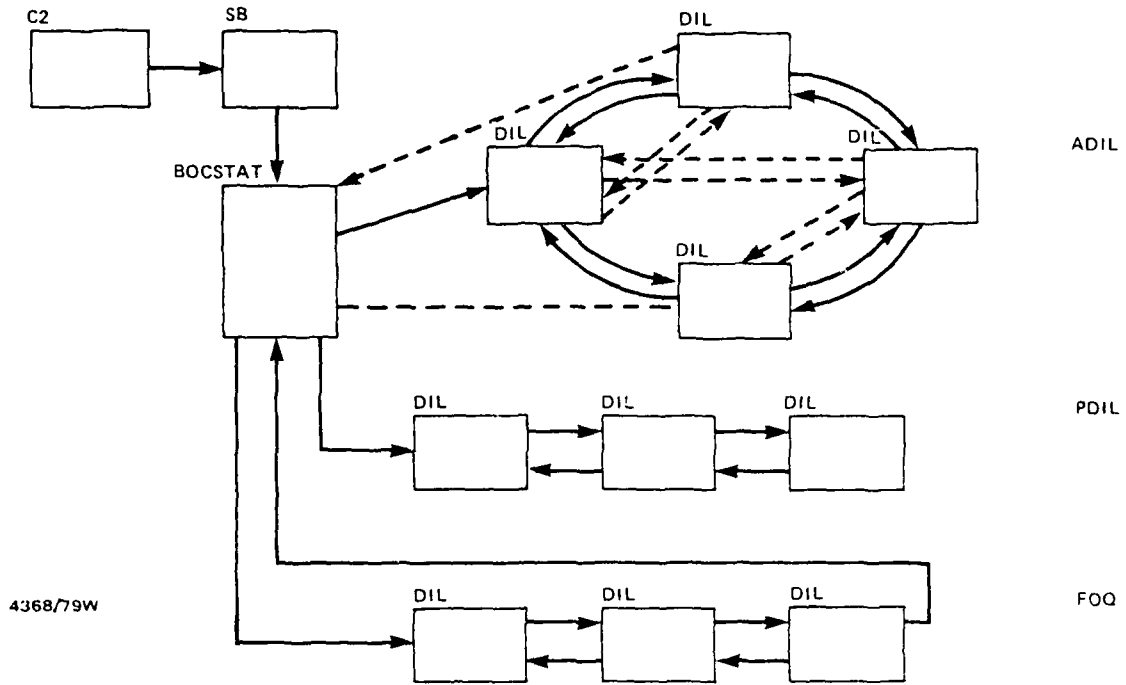


Figure III-28. Digested Information List Configurations

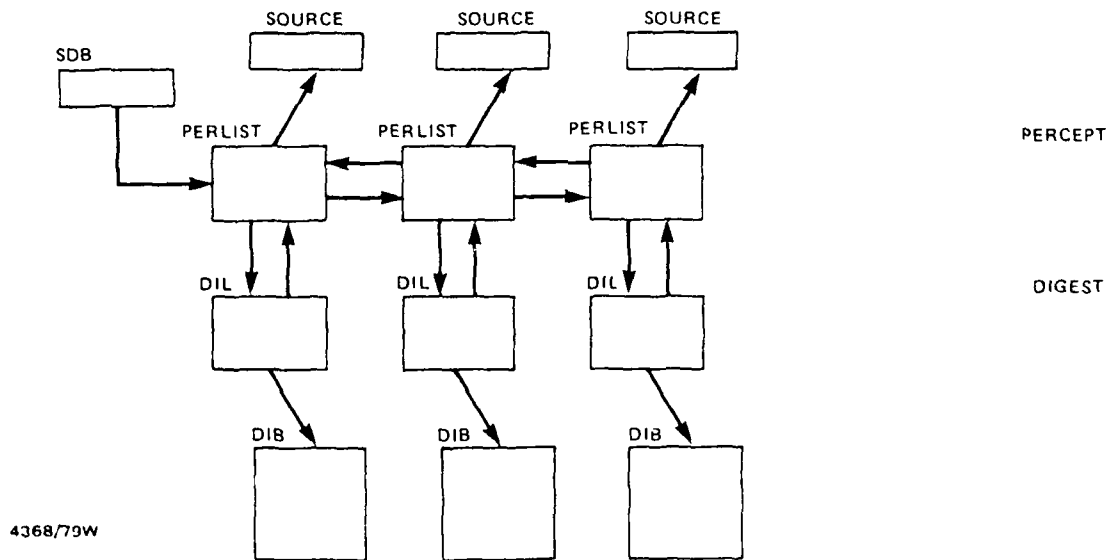


Figure III-29. Digested Information List and Perception List

SB	-	SCOREBOARD BLOCK. Contains HEX address; unit ID, PTR to BOCSTAT
SDB	-	STATUS DISPLAY BLOCK. Con- tains PTRS to subordinate list and perceptions list.
ADSITEDB	-	AIR DEFENSE SIGHTED DATA BLOCK. Contains information on the sighted aircraft
BOCSTAT	-	BOC STATUS BOARD. Contains PTR to the three DIL config- urations, ADIL, TRACKED, FOQ, along with general info on the BDC
DIL	-	DIGESTED INFORMATION LIST. Contains information on a target or possible target on coverage, engagement windows, and other info, contains PTRS to associated DIL and percep- tion block
SUBLIST	-	SUBORDINATE LIST BLOCK. Sub- ordinate to the BOC, can be associated with one or more targets
PAL	-	POSSIBILITIES AND ALLOCATIONS LIST. Created when a subor- dinate and a target are assoc- iated together
PERLIST	-	PERCEPTIONS LIST BLOCK. Forms a list, contains information on the seeing status of the target

DIB

DIGESTED INFORMATION BLOCK.

Contains information on the target as heading, velocity, and position

d. Block Specifications

1) Block Diagrams

a) C2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITYPE	SLIDE

b) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	ID
DATABASE	PBOCSTAT

c) SOB

PSB	PSEEBUF
SUBORDINATE	ORD

d) SUBLIST

BTRYSB	PNEXT
AUTO	NUMFU
RAMMO	LOAD
PDAQUE	NOUSE
PPAL	ADDRESS

e) BOCSTAT

AUTOFLG	READY
PSDIG	PSDELY
PADIL	NUMADIL
PHFOQ	PTFOQ
PHNDIL	PTNDIL
HPRIOR	TPRIOR
PPDIL	NUMROY
PHADAQ	PTDAQ

f) DIL

PUP	PDOWN
PPERCL	PDIB
LDC	PRIORITY
PRIUP	PRIDN
DCOV	SHORT
PDELAY	PENGEV
PPAL	PSUB

g) PAL

PUPT	PDOWNT
PUPB	PDOWNB
START (SPACE)	
END (SPACE)	
COVER	POIL
PEVENT	PSUB

h) ADSITEDB

PNEXT	AATYPE
MODVAL1	MAXNUMDIGEST
MAXTIMEDIGEST	MINTIMEDIGEST
LOSTTIME (SPACE)	
LASTCHANGE (SPACE)	
ENGAGEWINDOW(SPACE)	
MODVAL2 (SPACE)	
MODVAL3 (SPACE)	
LOVONONE	ONE
COVONFEW	FEW
COVONMANY	MANY
TIMEFLIGHT (SPACE)	
MISSILERANGE (SPACE)	
MAXASSIGN	MODVAL4
MODVAL5 (SPACE)	
MAXTRACKRANGE (SPACE)	
LOCKONTIME (SPACE)	

h) ADSITEDB (Continued)

MODVAL6 (SPACE)	
MODVAL7 (SPACE)	
CONVLOAD	
SNUKELoad	LNUKELoad
RESUPPLYCV	CVRESUPPLYFREQ
RESUPPLYSN	SNRESUPPLYFREQ
RESUPPLYLN	LNRESUPPLYFREQ

i) DIB

TIME (SPACE)	
SIDE	NUMAL
LOST	POSITION
HEADING (SPACE)	
VELOCITY (SPACE)	
ALTITUDE (SPACE)	

j) PERLIST

PUP	PDOWN
PSB	PDIL
SEEN	PSS
PUPCHN	PDNCHN
TIME (SPACE)	

k) SOURCE

PSB	PNEXT
-----	-------

2) Field Definitions

a) C² Block

- UNITNUMBER - Number of the unit. If negative, the unit is a passing target.
- PUP - Pointer to the C2 block of the unit's commander.
- PDOWN - Pointer to the C2 block of the unit's subordinate.

- PNEXT - Pointer to the C2 block of the unit's sibling.
 - PSB - Pointer to the SB block of the unit.
 - UNITTYPE - The unit's type code (see subsection f).
 - SIDE - Unit affiliation.
1 = Blue (NATO)
2 = Red (Pact)
- b) SB Block
- ADDRESS - Pointer to HEX block of the hex in which the unit is located.
 - PC2 - Pointer to C2 block of the unit.
 - PSDB - Pointer to the SDB block of the unit.
 - PFEL - Pointer to future event list EVENT block.
 - PACQ - Pointer to ACQBUF block. Used by CRC's for acquisition devices.
 - ID - Identification number.
 - DATABASE - If BOC or BTRY points to ADSITEDB block.
 - PBOCSTAT - Points to BOCSTAT block if the unit is a Battalion Operations Center.
- c) SDB Block
- PSB - Pointer to the unit's SB block.
 - PSEEBUF - Pointer to SEEBUF block which is used by aircraft flight units to record targets seen and their perceived damage levels. This field definition is used only by flights.

	SUBORDINATE	-	Points to <u>SUBLIST</u> block which is a list of subordinates.
	ORD	-	Points to <u>ORDERS</u> block if this unit is a flight.
d)	<u>SUBLIST</u>		
	BTRYSB	-	Pointer to the <u>SB</u> block of the BOC.
	PNEXT	-	Pointer to the <u>SUBLIST</u> block of the subordinate's sibling.
	AUTO	-	
	NUMFU	-	
	RAMMO	-	
	LOAD	-	
	PDAQUE	-	Pointer to the <u>DAQUE</u> block.
	NOUSE	-	
	PPAL	-	Pointer to the possibilities and allocations list block.
	ADDRESS	-	Pointer to the <u>HEX</u> block.
e)	<u>BOCSTAT</u>		
	AUTOFLG	-	
	READY	-	
	PSDIG	-	Pointer to the <u>EVENT</u> block which is the next scheduled digest event.
	PSDELY	-	Pointer to the <u>EVENT</u> block.
	PADIL	-	Pointer to the <u>DIL</u> block which is the next threat (DIL) to look at in the active DIL list.
	NUMADIL	-	Number of DILS in the active DIL list.
	PHFOQ	-	Pointer to the <u>DIL</u> block which is the head of the Force Out Queue.

- PTFOQ - Pointer to the DIL block which is the tail of the Force Out Queue.
- PHNDIL - Pointer to the PERLIST block. This pointer is not actively used in MADEM at present.
- PTNDIL - Pointer to the PERLIST block. This pointer is not actively used in MADEM at present.
- HPRIOR - Pointer to the DIL block which is the head of the priority chain within the active DIL list.
- TPRIOR - Pointer to the DIL block which is the tail of the priority chain within the active DIL list.
- PPDIL - Pointer to the DIL block which is the head of the passive DIL list.
- NUMROY - -
- PHDAQ - -
- PTDAQ - -
- f) DIL
- PUP - Pointer to the DIL block which is the pointer up in DIL chain.
- PDOWN - Pointer to the DIL block which is the pointer next in DIL chain.
- PPERCL - Pointer to the PERLIST block which is the pointer to the perceptions list.

PDIB - Pointer to the DIB block which is the digested information block. It holds the information about the flight's track.

LOC - The code to tell which DIL configuration this DIL is located.
 Code: 0 - Active Dil List
 1 - Force Out Queue
 2 - Passive Dil List

PRIORITY - Priority of

PRIUP - Pointer to the DIL block which is the up pointer in the priority chain in the active DIL list.

PDOWN - Pointer to the DIL block which is the down pointer in the priority chain in the active DIL list.

DCOV - Desired coverage of this target flight.

SHORT - Shortfall from desired coverage, i.e., Desired-Allocated = Shortfall.

PDELAY - Pointer to the DAQE block which points to the delayed action queue, a chain of delayed actions regarding this target.

PENGEV - Pointer to the EVENT block.

PPAL - Pointer to the PAL block which is the possibilities and allocations list block, a part of the threat-defense analysis.

PSUB - Pointer to the SUBLIST block.

- g) PAL
- PUPT - Pointer to the PAL block.
 - PDOWNT - Pointer to the PAL block.
 - PUPB - Pointer to the PAL block.
 - PDOWNB - Pointer to the PAL block.
 - START - Beginning time that the target is active for this BOC.
 - END - Ending time that the target is active for this BOC.
 - COVER - Amount of coverage the subordinate is giving this target.
 - PDIL - Pointer to the PAL block which is the target (DIL) associated with this PAL.
 - PEVENT - Pointer to the EVENT block.
 - PSUB - Pointer to the SUBLIST block which is the subordinate associated with this PAL.
- h) ADSITEDB Block
- PNEXT - Pointer to next ADSITEDB block.
 - AOTYPE - Unit type of this unit. Must be a BOC or BTRY.
 - MODVAL 1 - Model Value - 1.
 - MAXNUM DIGEST - Maximum number of flights on which a BOC or BTRY can be digesting info at one time.
 - MAX TIME DIGEST- Maximum time (in seconds) between consecutive digests of info (BOC and BTRY).
 - MIN TIME DIGEST- Minimum time (in seconds) between consecutive digests of info (BOC and BTRY).

LOST TIME - Time (in seconds) after which a track not seen is assumed permanently lost (BOC and BTRY).

LAST CHANCE - Time (in seconds) considered short for a subordinate to respond to a target. (Time from now until his last chance to shoot.) BOC only, for BTRY = 0.

ENGAGE WINDOW - Minimum length of subordinates engagement window for a significant engagement opportunity in seconds (BOC and BTRY).

MODVAL 2 - Model Value = 0

MODVAL 3 - Model Value = 0

COVONONE - Desired number of fire units coverage for one aircraft (BOC and BTRY).

ONE - Model Value = 1

COVONFEW - Desired number of fire units coverage for few aircraft (BOC and BTRY).

FEW - Model Value = 5, number of aircraft considered "few."

COVONMANY - Desired number of fire units coverage for many aircraft (BOC and BTRY).

MANY - Model Value = 1000000.

TIMEFLIGHT - Maximum time (in seconds) of flight for missile (BOC and BTRY).

MISSILE RANGE - Maximum range for missiles in meters (BOC and BTRY).

MAXASSIGN	-	Maximum number of targets per ready fire unit to be assigned at one time. BOC only, for BTRY = 0.
MODVAL 4	-	Model Value; for BOC = 8, BTRY = 11.
MODVAL 5	-	Model Value = 0
MAX TRACK RANGE-		Maximum tracking range in meters. BTRY only, for BOC = 0.
LOCK ON TIME	-	Assumed time (in seconds) for BTRY achieve lockon. BTRY only, for BOC = 0.
MODVAL 6	-	Model Value = 0.
MODVAL 7	-	Model Value = 0.
CONVLOAD	-	Number of conventional missiles.
SNUKELoad	-	Number of small nukes.
LNUKELoad	-	Number of large nukes.
RESUPPLYCV	-	Number of missiles per resupply of ammo. BTRY only, for BOC = 0.
CVRESUPPLYFREQ	-	Time (in seconds) between resupply of conventional ammo. BTRY only, for BOC = 0.
RESUPPLYSN	-	Number of missiles per resupply of small nukes. BTRY only, BOC = 0.
SNRESUPPLYFREQ	-	Time (in seconds) between resupply of small nukes. BTRY only, BOC = 0.
RESUPPLYLN	-	Number of missiles per resupply of large nukes. (BTRY only, BOC = 0).

LNRESUPPLYFREQ - Time (in seconds) between resupply of large nukes. BTRY only, BOC = 0.

i) DIB

TIME - Side of target (DIL).

NUMAL -

LOST - Code when target is lost.

POSITION - Positon of target.

HEADING - Heading of target (DIL).

VELOCITY - Velocity of target.

ALTITUDE - Altitude of target.

j) PERLIST

PUP - Pointer to the PERLIST block which is an up pointer in the perceptions list.

PDOWN - Pointer to the PERLIST block which is a next pointer in the perceptions list.

PSB - Pointer to the BOC's scoreboard.

PDIL - Pointer to the DIL block which is associated with it.

SEEN - Code to show seeing status of the target.

0 - Can not see and has not been assigned.

1 - Early warning, has been assigned by superior.

2 - Can see target.

PSS - Pointer to the SOURCE block which is a list of subordinates that can see the target.

PUPCHN -

PDNCHN -

- TIME - Time that the target was last seen.
- k) SOURCE
- PSB - Pointer to the SCOREBOARD block which is the scoreboard of the subordinate that can see the target.
- PNEXT - Pointer to the SOURCE block which is the next subordinate in the list that can see the target.

e. Linkages to Other Data Structures

The BOC structures are linked to the command and control tree by the C2 block of the BOC.

f. Notes

7. SAM BATTERY (BTRY)

a. Data Block Index

ABSITEDB

ALLOCATE

BTRYDIL

BTRYSTAT

C²

DIB

FIREUNT

PATENGAGE

PERLIST

SB

SDB

b. Description

The air defense information for the SAM Battery (BTRY) in MADEM is stored in the Battery Digested Information List (BTRYDIL). The fireunits of a Battery are allocated to the BTRYDILS to create the Threat-Defense Allocation Structure for each Battery.

The BTRYDIL represents the perceived threat to the Battery and is pointed to from the Battery's Status Board (BTRYSTAT). The list of fireunits of the Battery are pointed to from the Battery Status Display Board (SDB). The fireunits are the Battery's subordinates and the number of fireunits are limited depending on the type of Battery it is. A Hawk battery can have at most two fireunits where a Herc battery can have only one. When a fireunit is allocated to a threat, BTRYDIL, an ALLOCATE block is created to link the two together. Note that at most there can only be one allocation block for any one fireunit, but a threat, BTRDIL, could have multiple allocation blocks. This does not hold for the last Battery type, the PATRIOT. A PATRIOT does not have any fireunits as subordinates, it has a known engagement capacity. The number of engagements can be allocated to one or multiple threats, BTRYDILS. When a PATRIOT engagement is allocated to a threat a PATENGAGE block is created.

The BTRYDILS are divided up into three different DIL configurations; (1) Active DIL list, (2) TRACKED DIL list, (3) Force out queue. This allows the threats, DILS, to be handled in a systematic way. All the DIL configurations are pointed to from the BTRYSTAT.

The Active DIL (ADIL) is a doublylinked circular list of all the threats actively being considered by the Battery. The DIL's in the ADIL are considered one at a time in the list, in increments controlled by the Scheduled Digest event. The Air Defense Data Base controls the maximum number of DIL's a Battery can have in it's ADIL.

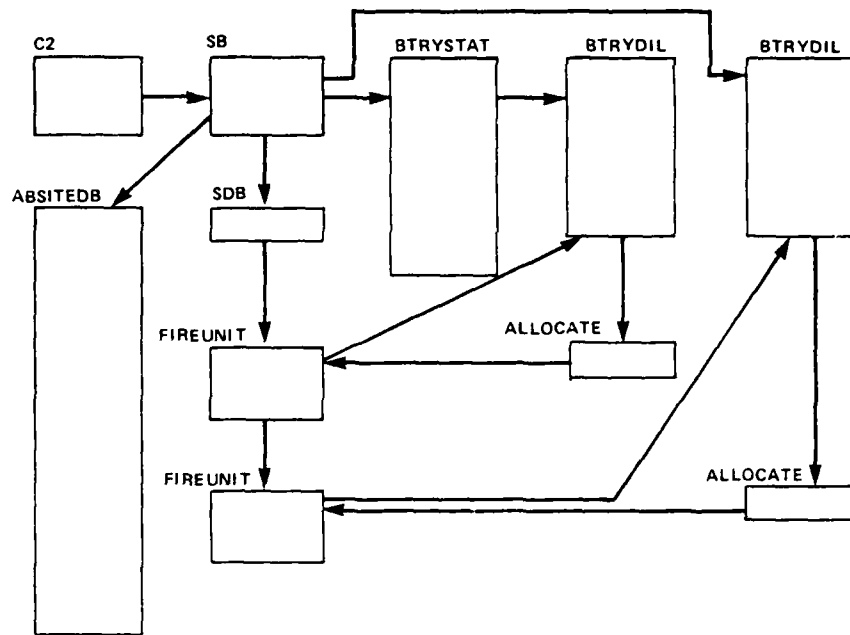
When a Battery fully assigns the responsibility of a threat DIL to it's subordinates, the DIL is taken out of the ADIL and put into the TRACKED DIL list. The Tracked List is a doubly linked list of DIL blocks with a pointer to the head of the list.

If a DIL in the ADIL is forced out because the need to bring in a DIL of a higher priority, it is put in the Force Out Queue (FOQ). The FOQ is a doubly linked list with a head and tail pointer to the list. This is an overflow list and the DIL will eventually be brought back into the ADIL when there is an opening. The head and tail pointers allow access from either end of the list to get the best DIL out of the FOQ.

Each DIL has an associated Digested Information Block (DIB) with it. This holds the current information that the BOC has on the target. When the DIL is in the Active DIL List a new DIB is created and compared to the old one each time the DIL target is examined. The DIL's are also associated with the Perceptions List, a PERLIST block from the Perceptions List and a DIL are linked up. The information in the PERLIST is from the Perception event and is used with the DIL to represent the information on a target.

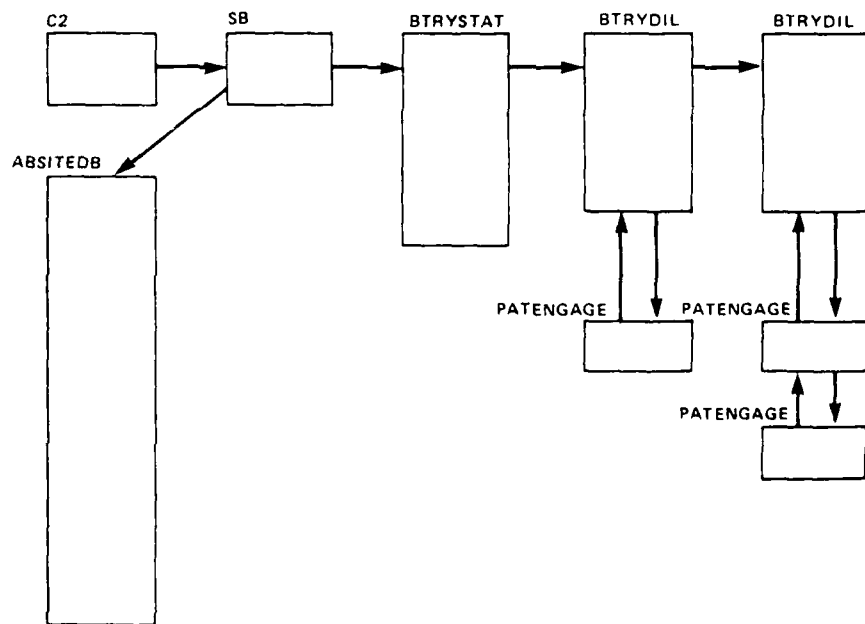
c. Structure Overview

1. Structure Diagram (Figure III-30, III-31, III-32, III-33)



4368/79W

Figure III-30. SAM Battery Structure Diagram (HAWK and HERC)



4368/79W

Figure III-31. SAM Battery Structure Diagram (PATRIOT)

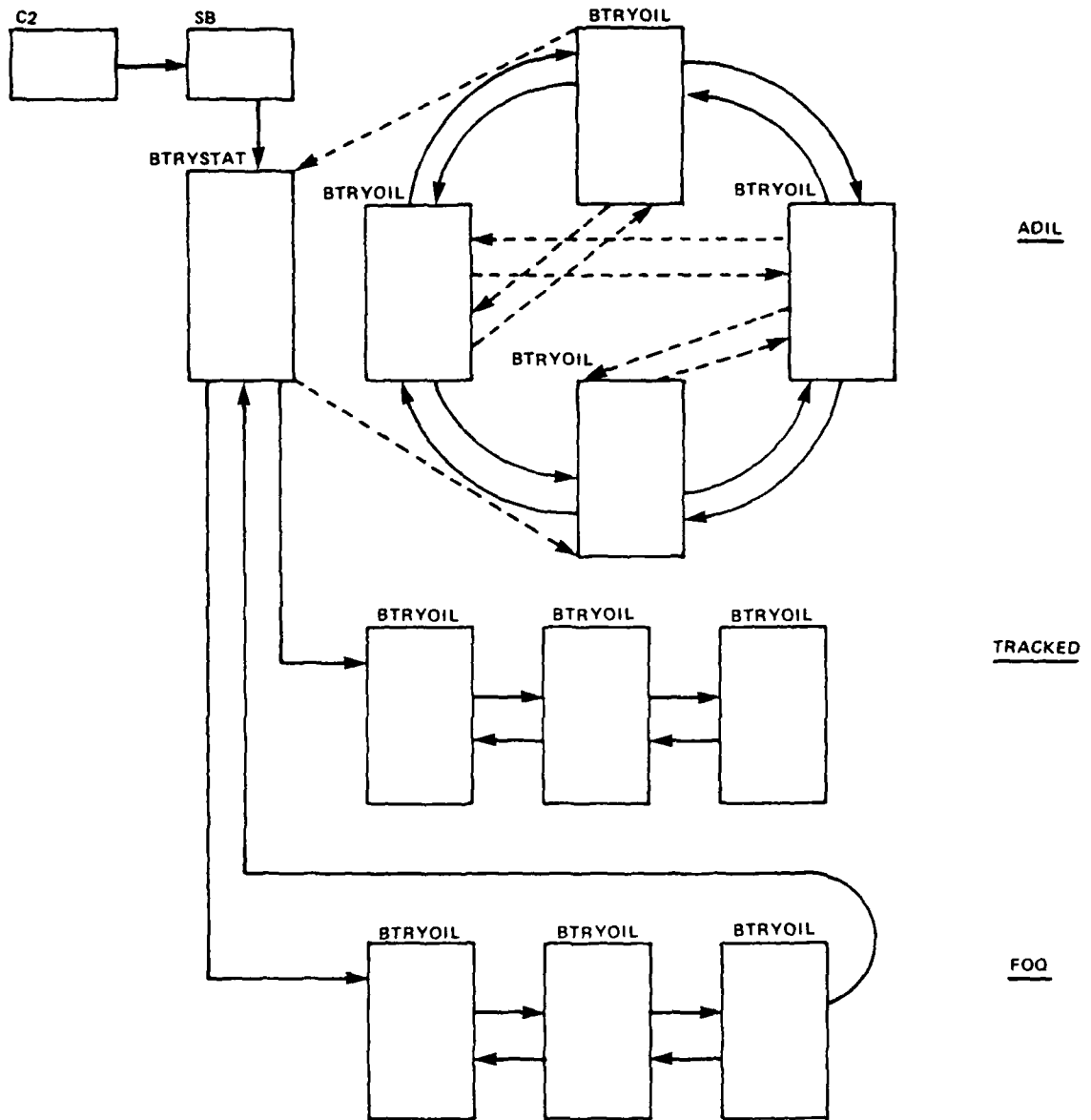
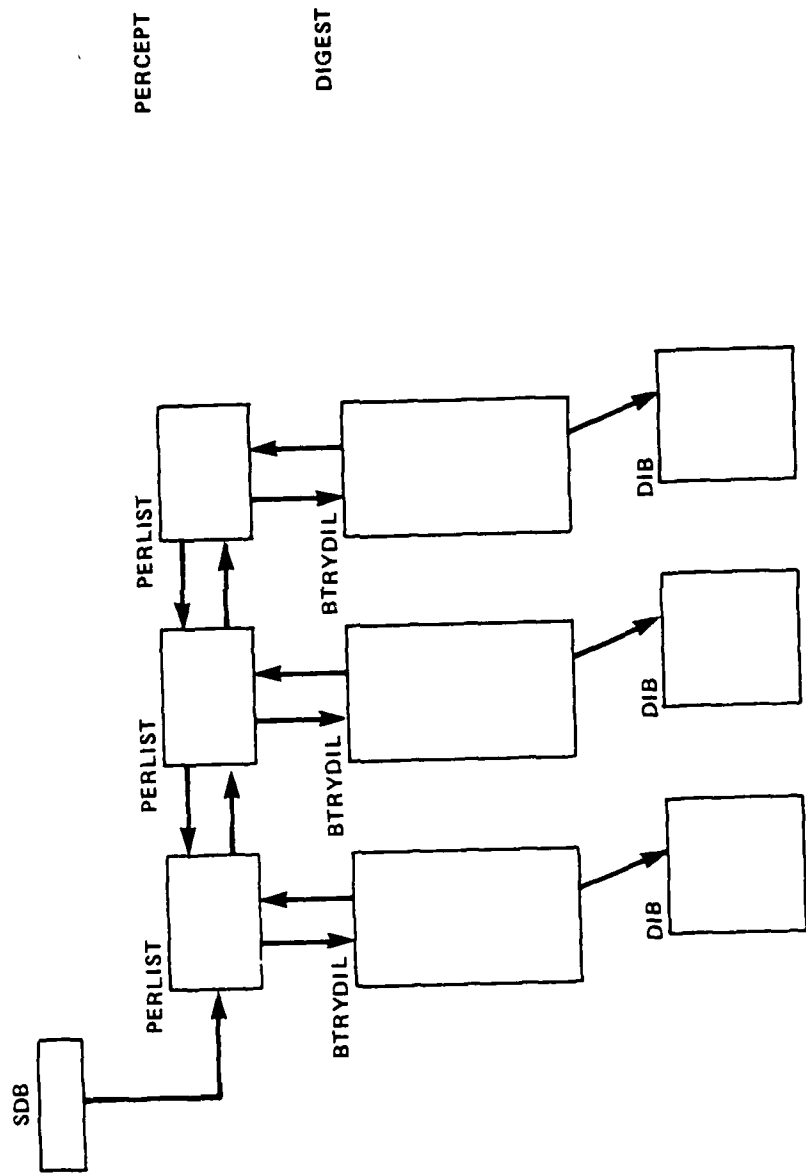


Figure III-32. BTRYDIL Configurations

4368/79W



PERCEPT

DIGEST

Figure III-33. Battery BTRYDIL Perception List Association

4368/79W

d. Block Specifications

1) Block Diagrams

a) c^2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

b) SB

ADDRESS	PC2
PSDB	PFEL
PACQ	IO
DATABASE	PBOCSTAT

c) SDB

PSB	PSEEBUF
SUBORDINATE	ORD

d) FIREUNIT

AMMO	PNEXT
PDIL	STAGE
CEASE	PENGEV
CAMMO	BUSV
NIAMMO	N2AMMO

e) BTRYSTAT

AUTO	NUM
PSDIG	PSDELY
PADIL	NADIL
PHFOQ	PTFOQ
PHNDIL	PTNDIL
HPRIOR	IPRIOR
TRACTED	IDLE
PHDAQ	PTDAQ
PTL	A1
RESUPPLY	A2
AMTDT	NUMENG
NUCNO	A3

f) BTRYDIL

PUP	PDOWN
PPERC	PDIB
LOC	PRIORITY
PRIUP	PRIDN
DCOV	SHORT
PDELAY	PENGEV
PAL	NUKE
CEASE	ASSIGN
TRACK	FIRE
START (SPACE)	
END (SPACE)	

g) ALLOCATE

PUP	PPDOWN
AMMOTYPE	PFU

h) ADSITEDB

PNEXT	ADTYPE
MODVAL1*	MAXNUMDIGEST
MAXTIMEDIGEST	MINTIMEDIGEST
LOSTTIME (SPACE)	
ENGAGEWINDOW (SPACE)	
MODVAL2 (SPACE)	
MODVAL3 (SPACE)	
COVONONE	ONE
COVONFEW	FEW
COVONMANY	MANY
TIMEFLIGHT (SPACE)	
MISSILE RANGE (SPACE)	
MAXASSIGN MODVAL4	
MODVAL5 (SPACE)	
MAXTRACKRANGE (SPACE)	
LOCKONTIME (SPACE)	

MODVAL6 (SPACE)	
MODVAL7 (SPACE)	
CONVLOAD	
SNUKELoad	LNUKELoad
RESUPPLYCV	CVRESUPPLYFREQ
RESUPPLYSN	SNRESUPPLYFREQ
RESUPPLYLN	LNRESUPPLYFREQ

i) PATENGAGE

PUP	PDOWN
PDIL	STAGE
CEASE	PENGEV

j) DIB

TIME(SPACE)	
SIDE	NUMAL
LOST	POSITION
HEADING (SPACE)	
VELOCITY (SPACE)	
ALTITUDE (SPACE)	

k) PERLIST

PUP	PDOWN
PSB	PDIL
SEEN	PSS
PULPCHN	PDNCHN
TIME (SPACE)	

2) Field Definitions

a) C2 Block

- UNITNUMBER - Number of the unit. If negative the unit is a passing target.
- PUP - Pointer to the C² block of the unit's commander.
- PDOWN - Pointer to the C² block of the unit's subordinate.

PNEXT	-	Pointer to the C ² block of the unit's sibling.
PSB	-	Pointer to the SB block of the unit.
UNITTYPE	-	The unit's type code (see subsection f)
SIDE	-	Unit affiliation. 1 = Blue (NATO) 2 = Red (PACT)
b)		
<u>SB BLOCK</u>		
ADDRESS	-	Pointer to HEX block of the HEX in which the unit is located.
PC2	-	Pointer to C ² block of the unit.
PSDB	-	Pointer to the SDB block of the unit.
PFEL	-	Pointer to future event list EVENT block.
PACQ	-	Pointer to ACQBUF block. Used by CRC's for acquisition devices.
ID	-	Identification number
DATABASE	-	If BOC or BTRY points to ADSITEDB block.
PBOCSTAT	-	Points to BOCSTAT block if the unit is a battalion operations center.
c)		
<u>SDB BLOCK</u>		
PSB	-	Pointer to the Unit's SB block.
PSEEBUF	-	Pointer to SEEBUF block which is used by aircraft flight units to record targets seen and their perceived damage levels. This field definition is used only by flights.

	SUBORDINATE	-	Points to FIREUNIT block which is a list of subordinate fire units.
	ORD	-	Points to ORDERS block if the unit is a flight.
d)	AMMO	-	
	PNEXT	-	Pointer to the FIREUNIT block of the subordinate fireunits sibling.
	PDIL	-	Pointer to the BTRVDIL block associate with this battery fireunit.
	STAGE	-	
	CEASE	-	
	PENGEV	-	
	CAMMO	-	
	BUSY	-	
	NIAMMO	-	
	N2AMMO	-	
e)	BTRYSTAT		
f)	BTRYDIL		
	PUP	-	Pointer to the BTRYDIL block which is the previous entry in the DIL chain.
	PDOWN	-	Pointer to the BTRYDIL block which is next entry in the DIL chain.
	PPERC	-	Pointer to the PERLIST block which is the entry in the perceptions list for this DIL.

PDIB - Pointer to the DIB block which is the digested information block holding information about the target.

LOC - The code to tell which DIL configuration this DIL is located in
 Code: 0 - Active DIL list
 1 - Force out queue
 2 - Tracked list

PRIORITY - Priority of this flight as a target.

PRIUP - Pointer to the BTRYDIL block which is the head of the priority chain within the active DIL list.

PRIDN - Pointer to the BTRYDIL block which is the tail of the priority chain within the active DIL list.

DCOV - Desired coverage on this flight.

SHORT - Shortfall from desired coverage:
 short - DCOV (DESIRED COVERAGE) - Allocated.

PDELAY - Pointer to the DAQE block which is the delayed action queue, a list of delayed action for this track.

PENGEV - Pointer to the EVENT block

PAL - Pointer to the ALLOCATE block which is the allocations list for this tracked target, holds the list of all fire units allocated to a target.

NUKE	-	Flags target NUKE/NoNUKE.
CEASE	-	'cease' marker
ASSIGN	-	
TRACK	-	Flag: 1 - waiting to track 2 - untrackable 0 -
FIRE	-	Waiting to fire flag
START	-	Starting window
END	-	End window
g) <u>ALLOCATE</u>		
PUP	-	Pointer to the ALLOCATE block which is the previous entry in the list of allocations.
PDOWN	-	Pointer to the ALLOCATE block which is the next entry in the list of allocations.
AMMOTYPE	-	
PFU	-	Pointer to the FIREUNIT block which is the fire unit associated with this allocation.
h) <u>ADSITEDB BLOCK</u>		
PNEXT	-	Pointer to next ADSITEDB block.
ADTYPE	-	Unit type of this unit, must be a BOC or BTRY.
MODVAL 1	-	Model value = 1
MAXNUM DIGEST	-	Maximum number of flights on which a BOC or BTRY can be digesting info at one time.
MAXTIMEDIGEST	-	Maximum time (inseconds) between consecutive digests of into (BOC & BTRY).

MINTIMEDIGEST - Minimum time (in seconds) between consecutive digests of into (BOC & BTRY)
LOSTTIME: - Time (in seconds) after which a track not seen is assumed permanently lost (BOC & BTRY).
LASTCHANCE - Time (in seconds) considered short for subordinate to respond to a target.
(time from now until his last chance to shoot). BOC only, for BTRY = 0.
ENGAGE WINDOW - Minimum length of subordinates engagement window for a significant engagement opportunity in seconds (BOC & BTRY).
MODVAL 2 - Model Value = 0
MODVAL 3 - Model value = 0
COVONONE - Desired number of fire units coverage for one aircraft (BOC & BTRY)
ONE - Model value = 1
COVONFEW - Desired number of fire units coverage for few aircraft (BOC & BTRY)
FEW - Model value = 5, number of aircraft considered "few".
COVONMANY - Desired number of fireunits coverage for many aircraft (BOC & BTRY).
MANY - Model value = 1000000.
TIMEFLIGHT - Maximum time (in seconds) of flight for missile (BOC & BTRY)

MISSILE RANGE - Maximum range for missiles in meters (BOC & BTRY).
 MAXASSIGN - Maximum number of targets per ready fire unit to be assigned at one time. BOC only, for BTRY = 0.
 MODVAL 4 - Model value; for BOC = 8, for BTRY = 11.
 MODVAL 5 - Model value = 0.
 MAX TRACK RANGE - Maximum tracking range in meters. BTRY only, for BOC = 0.
 LOCK ON TIME - Assumed time (in seconds) for BTRY to achieve lockon. BTRY only, for BOC = 0.
 MODVAL 6 - Model value = 0.
 MODVAL 7 - Model Value = 0.
 CONVLOAD - Number of conventional missiles.
 SNUKELoad - Number of large nukes.
 LNUKELoad - Number of large nukes.
 RESUPPLYCV - Number of missiles per resupply of ammo. BTRY only, for BOC = 0.
 CVRESUPPLYFREQ - Time (in seconds) between resupply of conventional ammo. BTRY only, for BOC = 0.
 RESUPPLYSN - Number of missiles per resupply of small nukes. BTRY only, BOC = 0.
 SNRESUPPLYFREQ - Time (in seconds) between resupply of small nukes. BTRY only, BOC = 0.
 RESUPPLYLN - Number of missiles per resupply of large nukes. (BTRY only, BOC = 0.

- LNRESUPPLYFREQ - Time (in seconds) between resupply of large nukes. BTRY only, BOC = 0.
- i) PATENGAGE
- PUP - Pointer to the PATENGAGE block which is the previous entry in the list of PATENGAGES below the BTRYDIL.
- PDOWN - Pointer to the PATENGAGE block which is the next entry in the list of PATENGAGES below the BTRYDIL.
- PDIL - Pointer to the BTRYDIL block associated with this PATENGAGE.
- STAGE -
- CEASE -
- PENGEV - Pointer to the EVENT block.
- j) DIB
- TIME -
- SIDE - Side of target (DIL)
- NUMAL -
- LOST - Code when target is lost.
- POSITION - Position of target
- HEADING - Heading of target (DIL)
- VELOCITY - Velocity of target
- ALTITUDE - Altitude of target
- k) PERLIST
- PUP - Pointer to the PERLIST block which is an up pointer in the perceptions list.
- PDOWN - Pointer to the PERLIST block which is a next pointer in the perceptions list.

PSB	-	Pointer to the DIL block which is associated with it.
SEEN	-	Code to show seeing status of the target 0 - can not see, and has not been assigned 1 - early warning, has been assigned by superior 2 - can see target
PSS	-	Not used with battery, only BOC
PUPCHN	-	
PDNCHN	-	
TIME	-	Time that the target was last seen

e. Linkages to other Data Structures

The battery structures are linked to the command and control tree by the C² block of the battery.

f. Notes

8. PASSIVE TARGETS (PT)

a. Data Block Index

C^2

SB (modified)

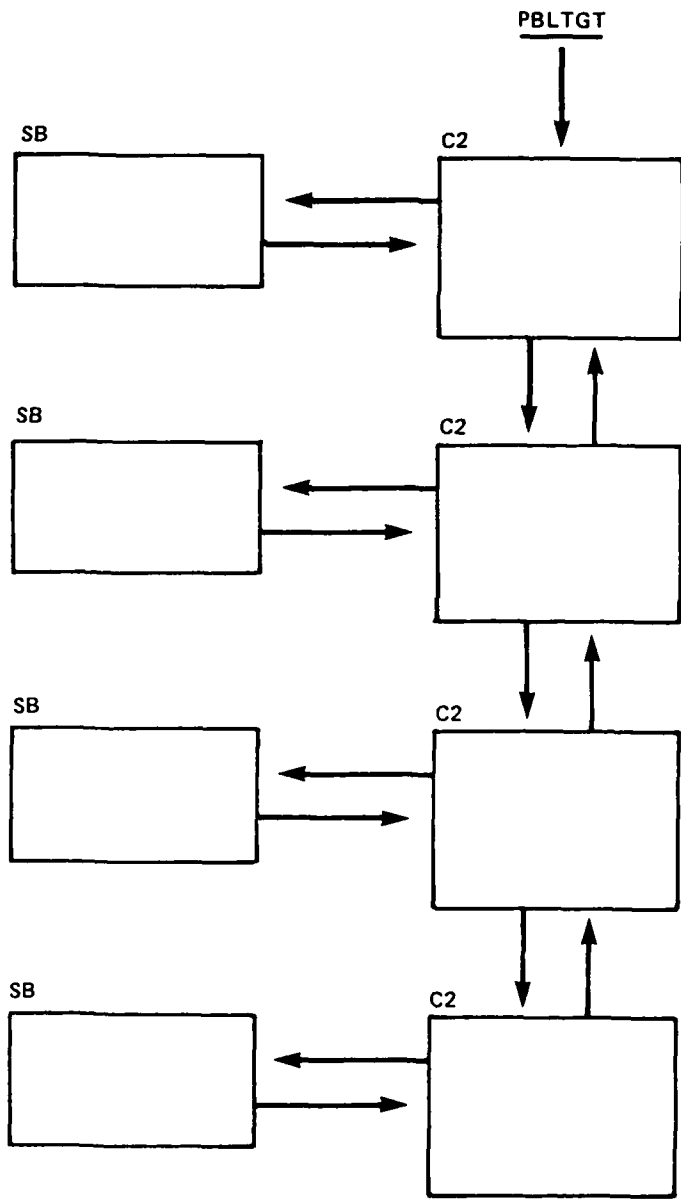
b. Description

The PASSIVE TARGET LIST is made up of C^2 and associated CB blocks which are in a linked list. This list is separate from the C^2 TREE and is used to represent blue units which are non-players. These blue target units are non-players in the sense that they merely act as objectives for red attacks.

Passive target units are characterized by negative unit numbers in their C^2 blocks and a modified SB block which has three rather than the usual four words. The configuration of the passive target list is shown in the structure diagram.

c. Structure Overview

1) Structure Diagram Figure III-34)



4368/79W

Figure III-34. Passive Target List Structure Diagram

2) Block Definitions

C^2

- COMMAND/CONTROL BLOCK. Contains list pointers, a negative unit number, pointer to SB, unit type code and side.
- The usual four words. Contains pointers to C^2 and HEX blocks.

d. Block Specifications

1) Block Diagrams

a) C^2

UNITNUMBER	
PUP	PDOWN
PSB	PNEXT
UNITTYPE	SIDE

b) SB (MODIFIED)

ADDRESS	PC2
CUMULTA IV DAMAGE	
PACQ	ID

2) Field Definitions

a) C^2 BLOCK

- UNITNUMBER - Number of the unit. If negative the unit is a passive target.
- PUP - Pointer to the C^2 block of the unit's commander.
- PDOWN - Pointer to the C^2 block of the unit's subordinate.
- PNEXT - Pointer to the C^2 block of the unit's sibling.
- PSB - Pointer to the SB block of the unit.
- UNITTYPE - The unit's type code (see subsection f)

SIDE - Unit affiliation.
1 = Blue (NATO)
2 = Red (PACT)

b) SB BLOCK

ADDRESS - Pointer to HEX block of the
hex in which the unit is located.
PC2 - Pointer to C² block of the unit.
CUMLTIVDAMAGE - Cumulative damage to target.
PACQ - Pointer to ACQBUF block. Used
by CRC's for acquisition
devices. (not used)
ID - Identification number.

e. Linkages to Other Data Structures

The PASSIVE TARGET LIST is used exclusively for targeting
by the red planning module. Its only external linkage is to the HEX block
in which the unit is located.

f. Notes

H. STORAGE SPACE MANAGEMENT

1. Data Block Index

Assorted unnamed unMIDASized blocks released from active use for data storage.

2. Description

Storage space in the dynamically allocated array ISPACE (131000) is carried out using the routines GIMME and RELEASE. GIMME is used to allocate storage blocks in ISPACE. Given the desired block length in words, GIMME first searches the Garbage Collection Matrix structure (illustrated in the structure diagrams) for a previously allocated but now released block of the desired length. If no block is found in the matrix, GIMME accesses NEW SPACE (Free Space) using the ESPTR pointer and allocates the desired block. If there is no space available, GIMME stops the simulation and issues a STORAGE SPACE OVERFLOW message.

RELEASE is used to construct the Garbage Collection Matrix. When a block is no longer required, GIMME is called with the length of the block to be released. RELEASE then adds the used block to the appropriate list in the matrix. The array ICTGIM(20) is used to count the number of released blocks in each size list. Blocks greater than or equal to 20 words in length are counted in element 20 of ICTGIM.

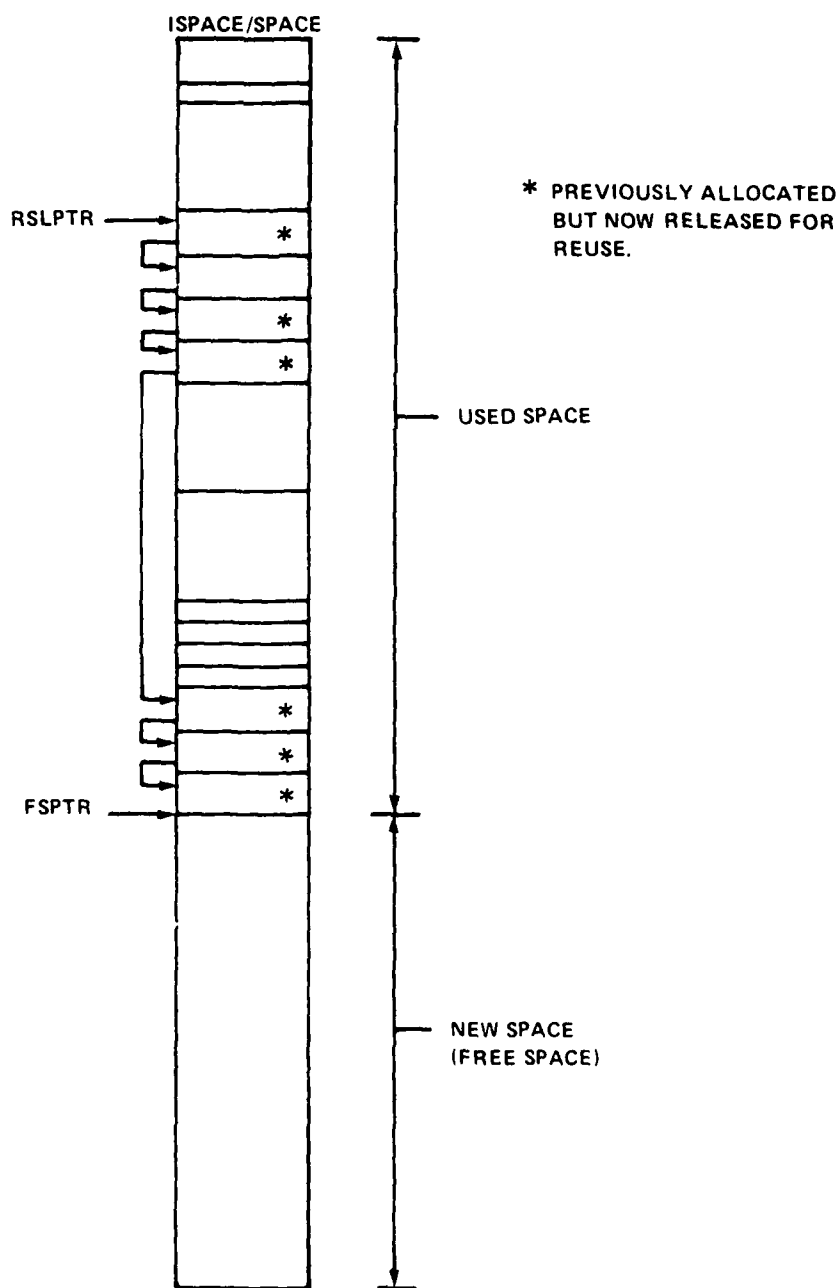
3. Structure Overview

a. Structure Diagrams (See Figures III-35 and III-36)

b. Block Definitions

BLKLENTH BLOCK SIZE LIST BUFFER BLOCK. Contains a block length field, a pointer to the list of BLK blocks of the specified length and a pointer to the next BLKLENTH block for the n next size SIZE LIST.

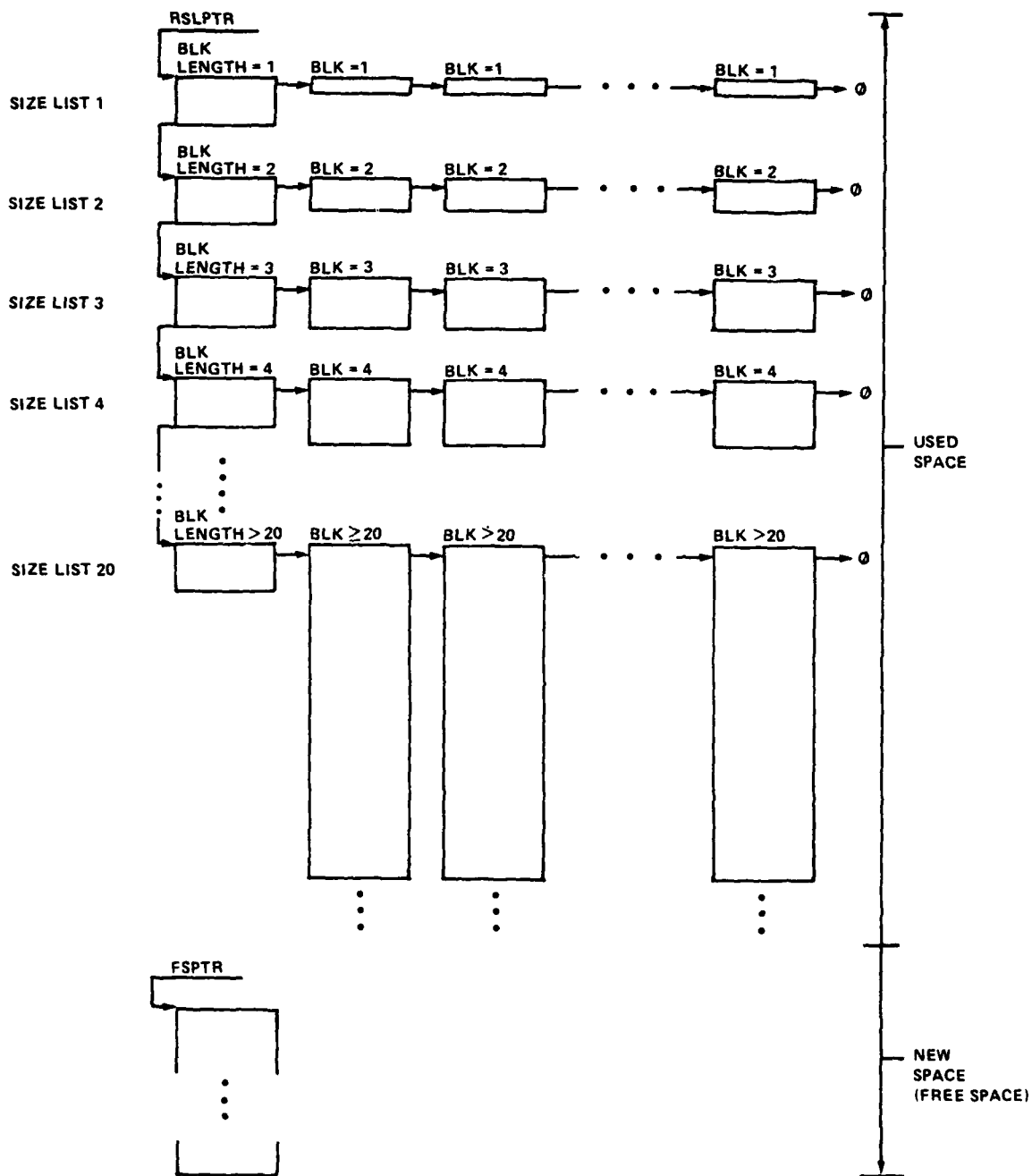
BLK RELEASED BLOCK. Storage block which has been released for further use. Contains a pointer to the next BLK block in its SIZE LIST. All remaining words initialized to zero (Ø).



RSLPTR - POINTER TO (ADDRESS OF) GARBAGE COLLECTION MATRIX
 FSPTR - POINTER TO (ADDRESS OF) FREE SPACE

4368/79W

Figure III-35. I-Space Array Configuration



4368/79W

Figure III-36. Garbage Collection Matrix Structure Diagram

4. Block Specifications

a. Block Diagrams

1) BLKLENGTH
BLOCKLENGTH
PTRBLOCK
PTRNEXT

2) BLK
PTRNEXT

Ø

Ø

Ø

Ø

VARIABLE LENGTH

o

o

o

b. Field Definitions

1) BLKLENGTH Block

BLOCKLENGTH Length of BLK blocks on list (in words).
PTRBLOCK Pointer to first BLK block in the list.
PTRNEXT Pointer to next BLKLENGTH block.

2) BLK Block

PTRNEXT Pointer to next BLK block in the size list.

5. Linkages to Other Data Structures

All other data structures are built using storage blocks from either the GARBAGE MATRIX or FREESPACE.

6. Notes

APPENDIX A
EXECUTION REQUIREMENTS

MADEM ON THE AFWL SYSTEM

MADEM uses either of 2 computers at AFWL, the Y mainframe (MFY) or the X mainframe (MFX). Both are CDC cyber 176 machines. We generally run on MFY since our data 100 has a direct line there, but we can run on either. It is possible to use a dial up terminal for either batch or interactive service using the following phone numbers:

<u>BAUD RATE</u>	<u>MFY</u>	<u>MFX</u>
300	505-264-2082 (3)	505-842-5162 (17)
300	--	505-842-9980 (10)
300	--	505-264-5875 (3)
300	--	505-265-9861 (3)
1200	505-264-5840 (3)	505-264-5705
1200	505-264-7812 (3)	--
4800	505-842-6392 (2)	505-842-6391 (4)
4800	--	505-842-5711 (6)

The number in parenthesis is the number of lines on that rotary.

USING THE DATA 100

The Data 100 (JB-5C) remote batch terminal has a direct line going to AFWL's MFY.

To Bring up the Data 100:

1. Load Emulator - Red in "Data 100" cards by pressing "HALT", then "LOAD". After cards have been read press "RUN".
2. Press Xmit button - Xmit light on is for 029 keypunch.
3. Wait for "Data Link" Light
4. Before entering each command, press control-A.
5. Type "LOGIN, SGCBDM, WDNA14V6, SUP (or L5 for V6)
6. Wait for "COMMAND" message.
7. Type "C".

To Enter Cards:

1. Load Cards
2. Press Start
3. Type "R" When Reader stops.

To Turn on Line Printer Type "ON,LP"

AFWL JOB CARD (1st Card in Deck)

Example: WDBMSM, ST176, T40, J037, P66.

WDBMSM - Can be any name, 1st 5 CHARS used as 1st 5 CHARS of 7 CHAR job name.

- ST176 - Tells it to run on either CYBER176 (MFY or MFX). Can also use STMFY or STMFY to run on a particular machine.
- T40 - CPU time limit in OCTAL seconds.
- I037 - IO time limit in OCTAL seconds.
- P66 - Request for 66 priority. The highest priority allowed is dependent on IO+CPU time requested.

P66 IO+CPU L 100 OCTAL
 P60 IO+CPU L 400 OCTAL
 P50 IO-CPU L 1,000 OCTAL

AFWL ACCOUNT CARD (2nd Card in Deck)

Example: Account BSMBDM, WDNA14V6-SGC, BDM, 703-821-4223.

- Account - Card Identifier
- BSMBDM - Means Nothing
- WDNA14V6 - Account
- SGC - Password
- BDM - Not needed, but it identifies us
- 703-821-4223 - Not needed, but is phone number of user in case the need to call user.

MADEM PERM FILES AT AFWL

<u>PERM FILE NAME</u>	<u>COMMENTS</u>
MADEM INITBIN	CURRENT INITBIN BINARY FILE
MADEM RUNBIN	CURRENT RUNBIN BINARY FILE
MADEM HISTBIN	CURRENT HISTBIN BINARY FILE
DATFILE AAA	DATFILE INPUT FOR AAA RUN
DATFILE TEST 1	DATFILE INPUT FOR TEST RUN
UOIL AAA	UOIL INPUT FOR AAA RUN
UOIL TEST 1	UOIL INPUT FOR TEST RUN
MIDASTABLE SOURCE	SOURCE FOR FULL MIOAS TABLE
MADEM IDAS TABLES	GENERATED MIDAS TBLES
ICOMPBIN	CURRENT BINARY FOR COMPARE 2ISPACES
NIPULSTOR	BINARY FOR NIPUL8TOR-DEBUG ROUTINE
KOMMONBIN	BINARY FOR COMMON ANALYZER
MADEM ONE PLAN 1 AAA	HOLD FILE LAST MADEM I INITBIN AAA RUN
MADEM ONE PLAN 2 AAA	HOLD FILE FROM CAST MADEM F AAA INITBIN RUN
MADEMONE INITBIN	MADEM I INITBIN BINARY, (UP TO M1)
MADEM ONE RUNBIN	MADEM I RUNBIN BINARY, (UP TO M1)
MADEM ONE HISTBIN	MADEM I RUNBIN BINARY, (UP TO M1)
MADEM TWORNITBIN	MADEM II INITBIN BINARY
MADEM TWO RUNBIN	MADEM II RUNBIN BINARY

LARGE PERM FILES AT AFWL

Any perm file larger than 35 RB's (1960 PRU's) will be routinely purged at AFWL.

1 PRU = 64 Words
1 RB = 56 PRU's = 3,584 Words
35 RB's = 125,440 Words

Theoretically, one can have a large perm file saved at AFWL if it is approved by Airman Vickers.

To have a perm file over 35 RB's saved write to:

Airman Richard Vickers
AFWL LADPO
Kirtland AFB
New Mexico 87117
(505) 264-7984

With the following information:

1. Justification for the large file
2. How long the file is to be saved
3. The name of the file
4. Your account number
5. The cycle numbers of that file to be protected.

To give a better probability to your request being approved send a cc to:

Pat Smari She works in the software consultine office and has
AFWL/ADSD promised to help us to get approval. Kirtland AFB
New Mexico 87117
(505) 264-0831

REGISTERING TAPES AT AFWL

AFWL Owned Tapes

To rent an AFWL owned tape simply call the AFWL Tape Librarian at:
(505) 264-0225

Have prepared the same information as is needed when sending a BDM tape (see below).

BDM Owned Tapes

If you don't have a tape, get one from Phuoy and send it to:

AFWL/ADPO
Kirtland ABF
New Mexico 87117
Attention: EXPEDITOR

Include the following information:

1. A 10-character tape name beginning with BDMV, C.I.E. BDMUMADEM 6)
2. The tape density
3. Number of tracks on tape
4. Your phone number (they will call with the USN)

MADEM TAPES

<u>VSN</u>	<u>TAPE NAME</u>	<u>TRACK</u>	<u>DENS</u>	<u>OWNED BY</u>	<u>LOCATION</u>	<u>DATE ACQUIRED</u>	<u>CONTAINS</u>
JH66		NT	HD	BDM	BDM		MADEM I SOURCE
JH67		NT	HD	BDM	BDM		MADEM I SOURCE
JH68		NT	HD	BDM	BDM		MADEM I PL
BC59		MT	HY	AFWL	AFWL		MADEM II PL
BC87		MT	HY	AFWL	AFWL		MADEL II PL
AB31	BDMUMADEM 1	MT	HY	AFWL	AFWL	7/79	MADEM III PL-BACKUP
AB33	BDMUMADEM 2	MT	HY	AFWL	AFWL	7/79	MADEM III PL-BACKUP
AD78	BDMUMADEM 3	MT	HY	BDM (24090)	AFWL	7/79	MADEM I PL-MOST RECENT 8/11/79
AD87	BDMUMADEM 4	MT	HY	BDM (24091)	AFWL	7/79	MADEM III PL-BACKUP
AI36	BDMUMADEM 5	MT	HY	AFWL	AFWL	8/79	MADEM III PL-CURRENT
AI76	BDMUMADEM 6	MT	HY	AFWL	AFWL	9/79	NOT USED.

NT = NINE TRACK HD = 800 CPI PL = PROGRAM LIBRARY
 MT = SEVEN TRACK HY = 800 BPI

AFWL ACCOUNTING
JULY 27, 1979

HRLY RATES

	<u>SYSTEM</u>	<u>CONNECT</u>
MFB	170/HR	\$5/HR, 7 A.M.-7 P.M.
MFX,	700/HR	\$10/HR, 7 A.M.-7 P.M.

BILLING BASED ON SYSTEM RECORDS
PERMFILE STORAGE FREE

BILLING QUESTIONS

Mr. Elsberrid (505) 264-0831
(CALL OR WRITE FOR DAY FILES)

AFWL/ADS
Attention: Consulting Office
Kirtland AFB
New Mexico 87117

In Particular call on accounting program errors for monthly bills call Dan Thornburg at (505) 264-0208.

HOW TO RUN MADEM

MADEM is currently set up to run on either the X or Y mainframe at AFWL.

To run MADEM you need on perm files:

1. UOIL (input data)
2. DATFILE (input data)
3. INITBIN (pre-processor binary file)
4. RUNBIN (main binary file)
5. HISTBIN (post-processor binary file)

MADEM is run in 3 stages; the middle stage is generally run in 6 to 12 steps, or volumes.

The 3 stages must be run in the following order: INITBIN, then RUNBIN once for each volume, then HISTBIN.

The complete run will result in 2 printouts plus one printout (the event trace) for each volume.

See Figure 1 for a run diagram, and the JCL listings for examples of JCL desks used for a run.

INITBIN

INITBIN is MADEM's pre-processor. It processes the DATFILE and UOIL, and then plans the red raid, all the while initializing 1 space and the common blocks. The 2 hold files save 1 space and the common blocks for subsequent volume runs. INITBIN also outputs a history file for the post-processor.

The Binary file INITBIN is stored on a perm file under the name 'MADEM-INITBIN' for MADEM III.

INITBIN INPUT:

1. DATFILE - Tape 8 - Perm File name is 'DATFILEAAA'
2. UOIL - Tape 7 - Perm File name is 'UOILAAA'
3. Input deck with parameters:
 - Card 1: (Unformatted, Mandatory)
 - Parm 1 - must be INTEGER 1 for INITBIN
 - Parm 2 - INTEGER, size of 1 space (max) during INITBIN
 - Parm 3 - Max number of players on either side
 - Parm 4 - Not used in INITBIN, max CPU time (real)
 - Card 2: To the last card:

The second set of options are all optional. Each of these parms must begin on column 10x+1, where x=(07). There may be from 1 to 8 RARMS per card, with as many cards as are necessary

Parms:

'DEBUG=ON'	-	Turns on printing or subroutine trace messages.
'DUMP=ON'	-	Will dump 1 space at end of run.
DATFILE=ON'	-	Turns on display of DATFILE DATA structure
'STOP=ODAT'	-	Stop INITBIN after DATFILE
'STOP=UOIL'	-	Stops INITBIN after semant (UOIL)
'STOP=DEL'	-	Stop INITBIN after DELADD, before executing plan event.

INITBIN OUTPUT:

1. 2 Hold files with 1 space, common blocks (Tape 10, Tape 11)
2. 1 File for history processing (Tape 4)
3. Printout, which includes:
 - 1) Plan of red raid
 - 2) Datfile structure pointers
 - 3) UOIL ECHOS:
 - commander-subordinate relationships
 - hex number for each unit, RED and BLUE
 - primary target line (PTL) values for each applicable BLUE unit
 - description of corridor block
 - hex and C² pointers for each unit, RED and BLUE

- 4) Common block dump
- 5) Other data, according to parameters, which may include:
 - subroutine trace messages
 - 1 space dump
 - data structure dumps

Perm Files Names for INITBIN output:

TAPE 10	-	PLAN 1 AAA
TAPE 11	-	PLAN 2 AAA
TAPE 4	-	HSTPLAN AAA

```

WDBMSM,ST176,T1177,I0177,P60,EC200.  MADEM INITBIN RUN (AAA)
ACCOUNT BSMBDM,WDNA14V6-SGC,BDM,703-821-4223.  B. MACALEER
COMMENT. *****
COMMENT. * *
COMMENT. * MADEM INITBIN RUN: *
COMMENT. * READS DATFILE, UOIL INPUT *
COMMENT. * GENERATES ISPACE *
COMMENT. * PLANS RED RAID *
COMMENT. * SAVES ISPACE ON HOLD FILES *
COMMENT. * SAVES COMMON BLOCKS ON HOLD FILES*
COMMENT. * *
COMMENT. * FILES: *
COMMENT. * TAPE4 - HISTORY FILE *
COMMENT. * TAPE6 - PRINTED OUTPUT *
COMMENT. * TAPE7 - UOIL INPUT *
COMMENT. * TAPE10 - FIRST HOLD FILE *
COMMENT. * TAPE11 - SECOND HOLD FILE *
COMMENT. * TAPE14 - PRINTED OUTPUT *
COMMENT. * *
COMMENT. *****
REQUEST,TAPE4,*PF.
REQUEST,TAPE10,*PF.
REQUEST,TAPE11,*PF.
ATTACH,TAPE7,UOIL78A,ID=WDNA14V6.
ATTACH,TAPE8,CDAT78A,ID=WDNA14V6.
ATTACH,XBIN,MADEMINITBIN,ID=WDNA14V6,MR=1.
LDSET,PRESET=ZERO,MAP=SBEX.
LOAD,XBIN.
EXECUTE.
DMP,100,7200.
REWIND,TAPE4.
REWIND,TAPE6.
REWIND,TAPE10.
REWIND,TAPE11.
REWIND,TAPE14.
COMMENT. *****
COMMENT. * SAVE HOLD FILES, HISTORY FILE *
COMMENT. *****
CATALOG,TAPE10,PLAN1AAA,ID=WDNA14V6,RP=999.
CATALOG,TAPE11,PLAN2AAA,ID=WDNA14V6,RP=999.
CATALOG,TAPE4,HSTPLANAAA,ID=WDNA14V6,RP=999.
COMMENT.
COPYBF,TAPE14,OUTPUT.
COPEBF,TAPE6,OUTPUT.
AUDIT.ID=WDNA14V6.
COMMENT. *****
COMMENT. * IF WE BOMBED, GET OUTPUT ANYWAY *
COMMENT. *****
EXIT.

```

```

DMP,100,7200.
REWIND,TAPE14.
COPYBF,TAPE14,OUTPUT.
REWIND,TAPE6.
COPYBF,TAPE6,OUTPUT.
COMMENT.
AUDIT,ID=WDNA14V6.
COMMENT. *****
COMMENT. * *
COMMENT. * FIRST INPUT CARD IS MANDATORY AND *
COMMENT. * HOLDS 4 PARAMETERS: *
COMMENT. * 1. IOP - MUST BE 1 FOR INITBIN *
COMMENT. * 2. MSPCE - SIZE OF ISPACE *
COMMENT. * 3. MAX NO. PLAYERS ON ONE SIDE *
COMMENT. * 4. MAX CPU TIME OF THIS RUN *
COMMENT. * *
COMMENT. * THE SECOND INPUT CARDS ARE OPTIONAL *
COMMENT. * AND CAN HOLD THESE PARMETERS: *
COMMENT. * DEBUT=ON - SUBROUTINE TRACE *
COMMENT. * MESSAGES ARE PRINTED *
COMMENT. * DUMP=ON - WILL DUMP ISPACE *
COMMENT. * DATFLE=ON - WILL DISPLAY DATFILE *
COMMENT. * STOP=ODAT - STOPS INITBIN AFTER *
COMMENT. * PROCESSING DATFILE,BEFORE UOIL *
COMMENT. * STOP=UOIL - STOPS AFTER UOIL *
COMMENT. * STOP=DEL - STOPS AFTER DELADD *
COMMENT. * *
COMMENT. *****
& EOR
1,50000,600,90.
DEBUG=ON
DUMP=ON
DATFLE=ON
STOP=ODAT
# EOI

```

RUNBIN

A full production run is generally accomplished in volumes, but may be done in one run if desired. The length of a volume is controlled by the fourth input parameter, which tells MADEM how much CPU time to use before stopping. Using 90 seconds, most MADEM runs take six to twelve volumes. The hold files between each volume are saved so that the user can rerun any given volume. This is done by changing the 'ATTACK' and 'CATALOG' cards (in the JCL deck) after each volume run. The hold files are numbered within the Perm File name to identify the volume that created them. For example, files AAV10, AAV11, AAV12, and AAV13 are hold files created by volume 1. The 'AAA' stands for run type AAA. There are currently input files on cards for 5 runs, called Runs AAA, BBB, CCC, DDD, and EEE.

Perm File Name for RUNBIN: 'MADEMRUNBIN'

RUNBIN INPUT:

1. 4 Hold files - TAPE15, TAPE16, TAPE17, TAPE18.
(For a Volume 1 Run use only TAPE15 and TAPE16).
2. Input deck with parameters:
Card 1:
 PARAM 1 - Must be INTEGER '2' for RUNBIN
 PARAM 2 - INTEGER, MAX size of ISPACE
 PARAM 3 - INTEGER, MAX number of players on either side.
 PARAM 4 - Real, MAX CPU time of volume (in seconds)
Card 2 to the last card:
 Same as INITBIN, but only DUMP=ON and DEBUG=ON are effective parameters.

RUNBIN OUTPUT:

1. 4 Hold files - TAPE10, TAPE11, TAPE12, TAPE13
2. TAPE4 - for the post processor
3. Event trace listing (printed)
4. Common block dump (printed)
5. DEBUG messages (if chosen)
6. ISPACE 'DUMP' (if chosen)

```

WDBMSM,ST176,T177,I0177,P60,EC400.  MADEM PRODUCTION RUN (AAA)
ACCOUNT BSMBDM,WDNA14V6-SGC,BDM,703-821-4223.  B.MACALEER
COMMENT. *****
COMMENT. * *
COMMENT. * MADEM PRODUCTION RUN *
COMMENT. * *
COMMENT. * FILES: *
COMMENT. * TAPE4 - HISTORY FILE *
COMMENT. * TAPE6 - PRINTED OUTPUT *
COMMENT. * TAPE10 - FIRST HOLD FILE *
COMMENT. * TAPE11 - SECOND HOLD FILE *
COMMENT. * TAPE12 - THIRD HOLD FILE *
COMMENT. * TAPE13 - FOURTH HOLD FILE *
COMMENT. * TAPE14 - PRINTED OUTPUT *
COMMENT. * *
COMMENT. *****
REQUEST,TAPE4*PF.
REQUEST,TAPE10,*PF.
REQUEST,TAPE11,*PF.
REQUEST,TAPE12,*PF.
REQUEST,TAPE13,*PF.
COMMENT.
ATTACK,BIN,MADEMRUNBIN,ID=WDNA14V6,MR=1.
LDSET,PRESET=ZERO.
LOAD,BIN.
EXECUTE,,PL=20000.
REWIND,TAPE4.
REWIND,TAPE6.
REWIND,TAPE10.
REWIND,TAPE11.
REWIND,TAPE12.
REWIND,TAPE13.
REWIND,TAPE14.
COMMENT. *****
COMMENT. *SAVE ISPACE, COMMONS *
COMMENT. *SAVE TAPE4 FOR HISTORY PROCESSING *
COMMENT. *****
CATALOG,TAPE10,AAAV10,ID=WDNA14V6,RP=999.
CATALOG,TAPE11,AAAV11,ID=WDNA14V6,RP=999.
CATALOG,TAPE12,AAAV12,ID=WDNA14V6,RP=999.
CATALOG,TAPE13,AAAV13,ID=WDNA14V6,RP=999.
CATALOG,TAPE4,HSIAAAV1,ID=WDNA14V6,RP=999.
COMMENT.
DMP,100,7200.
COMMENT. *****
COMMENT. *TAPE14 USUALLY NULL *
COMMENT. *****
COPYBF,TAPE14,OUTPUT.
COMMENT.

```

```

COMMENT. *****
COMMENT. *TAPE6 HAS EVENTS, DUMPS (IF ANY) *
COMMENT. *****
COPYBF,TAPE6,OUTPUT.
COMMENT.
AUDIT, ID=WDNA14V6.
COMMENT. *****
COMMENT. * IF WE BOMB, PRINT OUTPUT ANYWAY *
COMMENT. *****
EXIT.
DMP, 100, 7200.
REWIND, TAPE14.
COPYBF, TAPE14, OUTPUT.
REWIND, TAPE6.
COPYBE, TAPE6, OUTPUT.
AUDIT, ID=WDNA14V6.
COMMENT. *****
COMMENT. * *
COMMENT. * FIRST INPUT CARD IS MANDATORY AND *
COMMENT. * HOLDS 4 PARAMETERS: *
COMMENT. * 1. IOP - MUST BE 2 FOR RUNBIN *
COMMENT. * 2. MSPCE - SIZE OF ISPACE *
COMMENT. * 3. MAX NO. PLAYERS ON ONE SIDE *
COMMENT. * 4. MAC CPU TIME OF THIS RUN *
COMMENT. * *
COMMENT. * THE SECOND INPUT CARDS ARE OPTIONAL *
COMMENT. * AND CAN HOLD THESE PARAMETERS: *
COMMENT. * DEBUG=ON - SUBROUTINE TRACE *
COMMENT. * MESSAGES ARE PRINTED *
COMMENT. * DUMP=ON - WILL DUMP ISPACE *
COMMENT. * *
COMMENT. *****
& EOR
2, 131000, 600, 90.
DEBUG=ON
DUMP=ON
# E01

```

CARDS FOR VOLUMES:

COMMENT.
COMMENT. *****
CINEBT, * VOLUME 2, RUN TYPE AAA *
COMMENT. *****
COMMENT.
ATTACH,TAPE15,AAAV10,ID=WDNA14V6.
ATTACH,TAPE16,AAAV11,ID=WDNA14V6.
ATTACH,TAPE17,AAAV12,ID=WDNA14V6.
ATTACH,TAPE18,AAAV13,ID=WDNA14V6.
CATALOG,TAPE10,AAAV20,ID=WDNA14V6,RP=999.
CATALOG,TAPE11,AAAV21,ID=WDNA14V6,RP=999.
CATALOG,TAPE12,AAAV22,ID=WDNA14V6,RP=999.
CATALOG,TAPE13,AAAV23,ID=WDNA14V6,RP=999.
CATALOG,TAPE4,HSTAAAV2,ID=WDNA14V6,RP=999.
COMMENT.
COMMENT. *****
COMMENT. *VOLUME 3, RUN TYPE AAA *
COMMENT. *****
COMMENT.
ATTACH,TAPE15,AAAV20,ID=WDNA14V6.
ATTACH,TAPE16,AAAV21,ID=WDNA14V6.
ATTACH,TAPE17,AAAV22,ID=WDNA14V6.
ATTACH,TAPE18,AAAV23,ID=WDNA14V6.
CATALOG,TAPE10,AAAV30,ID=WDNA14V6,RP=999.
CATALOG,TAPE11,AAAV31,ID=WDNA14V6,RP=999.
CATALOG,TAPE12,AAAV32,ID=WDNA14V6,RP=999.
CATALOG,TAPE13,AAAV33,ID=WDNA14V6,RP=999.
CATALOG,TAPE4,HSTAAAV3,ID=WDNA14V6,RP=999.
COMMENT.
COMMENT. *****
COMMENT. *VOLUME 4, RUN TYPE AAA *
COMMENT. *****
COMMENT.
ATTACH,TAPE15,AAAV30,ID=WDNA14V6.
ATTACH,TAPE16,AAAV31,ID=WDNA14V6.
ATTACH,TAPE17,AAAV32,ID=WDNA14V6.
ATTACH,TAPE18,AAAV33,ID=WDNA14V6.
CATALOG,TAPE10,AAAV40,ID=WDNA14V6,RP=999.
CATALOG,TAPE11,AAAV41,ID=WDNA14V6,RP=999.
CATALOG,TAPE12,AAAV42,ID=WDNA14V6,RP=999.
CATALOG,TAPE13,AAAV43,ID=WDNA14V6,RP=999.
CATALOG,TAPE4,HSTAAAV4,ID=WDNA14V6,RP=999.
COMMENT.
COMMENT. *****
COMMENT. *VOLUME 5, RUN TYPE AAA *
COMMENT. *****
COMMENT.
ATTACH,TAPE15,AAAV40,ID=WDNA14V6.

ATTACH,TAPE16,AAAV41, ID=WDNA14V6.
ATTACH,TAPE17,AAAV42, ID=WDNA14V6.
ATTACH,TAPE18,AAAV43, ID=WDNA14V6.
CATALOG,TAPE10,AAAV50, ID=WDNA14V6, RP=999.
CATALOG,TAPE11,AAAV51, ID=WDNA14V6, RP=999.
CATALOG,TAPE12,AAAV52, ID=WDNA14V6, RP=999.
CATALOG,TAPE13,AAAV53, ID=WDNA14V6, RP=999.
CATALOG,TAPE4,HSTAAAV5, ID=WDNA14V6, RP=999.
COMMENT.
COMMENT. *****
COMMENT. * VOLUME 6, RUN TYPE AAA *
COMMENT. *****
COMMENT.
ATTACH.TAPE15,AAAV50, ID=WDNA14V6

HISTBIN

After all volumes have been run, the post processor can be run to summarize the outcome of events.

PERM FILE name of BINARY FILE: HISTBIN

INPUT:

- (1) TAPE4 - A concatenation of all post-processor (history) files which includes exactly one file for each volume plus one file from INITBIN
- (2) Two card input deck of PARMs, as in JCL example.

OUTPUT:

Printed summary of the MADEM RUN.

NOTE: Little is known about HISTBIN AT THIS TIME.

WDBMBSM,ST176,T177,10177,P60. MADEM HISTORY PROCESSING
ACCOUNT BSMBDM,WDNA14V6-SGC,BDM,703-821-4223. B.MACALEER
ATTACH,LGO,MADEMHISTBIN,ID=WDNA14V6,MR=1.
ATTACH,PLAN,HSTPLANAAA,ID=WDNA14V6.
ATTACH,V1,HSTAAAV1,ID=WDNA14V6.
ATTACH,V2,HSTAAAV2,ID=WDNA14V6.
ATTACH,V3,HSTAAAV3,ID=WDNA14V6.
ATTACH,V4,HSTAAAV4,ID=WDNA14V6.
ATTACH,V5,HSTAAAV5,ID=WDNA14V6.
ATTACH,V6,HSTAAAV6,ID=WDNA14V6.
ATTACH,V7,HSTAAAV7,ID=WDNA14V6.
ATTACH,V8,HSTAAAV8,ID=WDNA14V6.
ATTACH,V9,HSTAAAV9,ID=WDNA14V6.
COPYBR,PLAN,TAPE4.
COPYBR,V1,TAPE4.
COPYBR,V2,TAPE4.
COPYBR,V3,TAPE4.
COPYBR,V4,TAPE4.
COPYBR,V5,TAPE4.
COPYBR,V6,TAPE4.
COPYBR,V7,TAPE4.
COPYBR,V8,TAPE4.
COPYBR,V9,TAPE4.
REWIND,TAPE4.
LDSET,PRESET=ZERO.
LOAD,LGO.
EXECUTE.
REWIND,TAPE4.
COPYSBF,TAPE4,OUTPUT.
AUDIT,ID=WDNA14V6.
EXIT.
AUDIT,ID=WDNA14V6.
& EOR
CONVENTIONAL 1978 AAA LEVEL 2 PK
39000.,999999999.,999999999.,999999999.,999999999.
EOI

RUNBIN

A full production run is generally accomplished in volumes, but may be done in one run if desired. The length of a volume is controlled by the fourth input parameter, which tells MADEM how much CPU time to use before stopping. Using 90 seconds, most MADEM runs take six to twelve volumes. The hold files between each volume are saved so that the user can rerun any given volume. This is done by changing the 'ATTACK' and 'CATALOG' cards (in the JCL deck) after each volume run. The hold files are numbered within the Perm File name to identify the volume that created them. For example, files AAAV10, AAAV11, AAAV12, and AAAV13 are hold files created by volume 1. The 'AAA' stands for run type AAA. There are currently input files on cards for 5 runs, called Runs AAA, BBB, CCC, DDD, and EEE.

Perm File Name for RUNBIN: 'MADEM RUNBIN'

RUNBIN INPUT:

1. 4 Hold files - TAPE15, TAPE16, TAPE17, TAPE18.
(For a Volume 1 Run use only TAPE15 and TAPE16).
2. Input deck with parameters:
Card 1:
 PARAM 1 - Must be INTEGER '2' for RUNBIN
 PARAM 2 - INTEGER, MAX size of ISPACE
 PARAM 3 - INTEGER, MAX number of players on either side.
 PARAM 4 - Real, MAX CPU time of volume (in seconds)
Card 2 to the last card:
 Same as INITBIN, but only DUMP=ON and DEBUG=ON are effective parameters.

RUNBIN OUTPUT:

1. 4 Hold files - TAPE10, TAPE11, TAPE12, TAPE13
2. TAPE4 - for the post processor
3. Event trace listing (printed)
4. Common block dump (printed)
5. DEBUG messages (if chosen)
6. ISPACE 'DUMP' (if chosen)

WDBMSM,ST176,T177,I0177,P60,EC400. MADEM PRODUCTION RUN (AAA)
ACCOUNT BSMBDM,WDNA14V6-SGC,BDM,703-821-4223. B.MACALEER
AUDIT,ID=WDNA14V6.

COMMENT. *****

COMMENT. *****

COMMENT. * PURGE INITBIN'S OUTPUT FILES *

COMMENT. *****

COMMENT. *****

PURGE,PLAN1,PLAN1AAA,ID=WDNA14V6,LC=1.

PURGE,PLAN2,PLAN2AAA,ID=WDNA14V6,LC=1.

PURGE,HSTPLANAAA,ID=WDNA14V6,LC=1.

COMMENT. *****

COMMENT. *PURGE VOLUME 1 OUTPUT FILES *

COMMENT. *****

PURGE,AAAV10,ID=WDNA14V6,LC=1.

PURGE,AAAV11,ID=WDNA14V6,LC=1.

PURGE,AAAV12,ID=WDNA14V6,LC=1.

PURGE,AAAV13,ID=WDNA14V6,LC=1.

PURGE,HSTV1,HSTAAAV1,ID=WDNA14V6,LC=1.

COMMENT. *****

COMMENT. *PURGE VOLUME 2 OUTPUT FILES *

COMMENT. *****

PURGE,AAAV20,ID=WDNA14V6,LC=1.

PURGE,AAAV21,ID=WDNA14V6,LC=1.

PURGE,AAAV22,ID=WDNA14V6,LC=1.

PURGE,AAAV23,ID=WDNA14V6,LC=1.

PURGE,HSTV2,HSTAAAV2,ID=WDNA14V6,LC=1.

COMMENT. *****

COMMENT. *PURGE VOLUME 3 OUTPUT FILES *

COMMENT. *****

PURGE,AAAV30,ID=WDNA14V6,LC=1.

PURGE,AAAV31,ID=WDNA14V6,LC=1.

PURGE,AAAV32,ID=WDNA14V6,LC=1.

PURGE,AAAV33,ID=WDNA14V6,LC=1.

PURGE,HSTV3,HSTAAAV3,ID=WDNA14V6,LC=1.

COMMENT. *****

COMMENT. * PURGE VOLUME 4 OUTPUT FILES *

COMMENT. *****

PURGE,AAAV40,ID=WDNA14V6,LC=1.

PURGE,AAAV41,ID=WDNA14V6,LC=1.

PURGE,AAAV42,ID=WDNA14V6,LC=1.

PURGE,AAAV43,ID=WDNA14V6,LC=1.

PURGE,HSTV4,HSTAAAV4,ID=WDNA14V6,LC=1.

COMMENT. *****

COMMENT. * PURGE VOLUME 5 OUTPUT FILES *

COMMENT. * *****

PURGE,AAAV50,ID=WDNA14V6,LC=1.

PURGE,AAAV51,ID=WDNA14V6,LC=1.

PURGE,AAAV52,ID=WDNA14V6,LC=1.

```
PURGE,AAAV53,ID=WDNA14V6,LC=1.
PURGE,HSTV5,HSTAAAV5,ID=WDNA14V6,LC=1.
COMMENT. *****
COMMENT. * PURGE VOLUME 6 OUTPUT FILES *
PURGE,AAAV60,ID=WDNA14V6,LC=1.
PURGE,AAAV61,ID=WDNA14V6,LC=1.
PURGE,AAAV62,ID=WDNA14V6,LC=1.
PURGE,AAAV63,ID=WDNA14V6,LC=1.
PURGE,HSTV6,HSTAAAV6,ID=WDNA14V6,LC=1/
AUDIT,ID=WDNA14V6.
&      EOR
#      EOI
```

The MIDAS BIN code is called "MIDAS" on account BDMAFM at AFWL.

JCL: MIDAS, COMPILE, XXX. this run a file called 'COMPILE'
 OPTION: PL=nnn (generally the output from UPDATE)
 where nnn=line limit of through MIDAS and puts the compilable
 FTNOUT FILE. fortran code into a file XXX. The
 Default: nnn=5,000. default for XXX is FTNOUT.

MADEM MIDAS TABLES

- (1) "4" Data Structures
- (2) All Data Structures

PROGRAM LIBRARY DECK NAME CHANGES

The following subroutines were listed in decks on the program library whose deck name did not match the subroutine name. The deck names have been changed to match the subroutine names. In some MADEM I and MADEM II listings. However, these routines are filed under the old deck name.

<u>SUBROUTINE</u>	<u>OLD DECK NAME</u>	<u>NEW DECK NAME</u>
AB2CRC	ABMSG	AB2CRC
BNCONTC	NBNCONT	BNCONTC
BNLALLE	NBNLALL	BNLALLE
BTN2CRC	BTNMSG	BTN2CRC
BYALCOV	PROCCLU	BYALCOV
BYCONTC	NBYCONT	BYCONTC
BYTKCHK	NBYTKCH	BYTKCHK
CRC2INT	INTASIN	CRC2INT
HANDZPT	NHANDZP	HANDZPT
INRANGE	NINRANG	INRANGE
INTASIN	BADASIN	INTASIN
INT2CRC	INTMSG	INT2CRC
RELOCAT	NRELOC	RELOCAT
SAMPRCM	NSAMPRC	SAMPRCM
TRKCHEK	NTRKCHE	TRKCHEK

The only remaining subroutines that do not occupy a deck of the same name are the UOIL routines in decks ULSUB2 and UOILSUB.

STRAY SUBROUTINES

The following subroutines reside in the noted deck on the program library. These routines are standard UOIL routines, and need not be altered for MADEM. These routines are represented on our master list by deck name, and may be filed as such in the books.

DECK UOILSUB

APCEL1
APCEL2
CARD
CHRGEN
LACELL
NXTSYM

DECK ULSUB2

ADDCHR
ERROR
EXTSCN
LEXAN
LOOKUP
LRKPRS

APPENDIX B
MODULAR INFORMATION DATA ACCESS SYSTEM (MIDAS)

A. INTRODUCTION

MIDAS consists of two parts: the MIDAS language and the MIDAS Translator. The MIDAS language allows one to write programs using only the logical aspects of data structures. The MIDAS Translator reads data structure definitions and programs written in the MIDAS language, realizes the physical implementation of the logical data structures, and generates a FORTRAN program or subprogram as output.

The advantages of using MIDAS are twofold. First, the programmer is no longer concerned with the details of data structure implementation, and is free to use natural names for elements of the data structures. Second, a program written in MIDAS is easier to convert to another type of computer since the logical definition of the data structures do not change, only the physical implementation which is now completely mechanized.

The MIDAS Translator is controlled by a set of tables which define the logical data structures to be translated and their specific physical implementation. Thus, each programming project will utilize a different set of tables, corresponding to the data structures peculiar to that project.

The MIDAS Translator constructs these tables automatically using information supplied by the user. This definition information is expressed using a Data Structure Definition Language (DSDL) which defines the logical data structures and their physical implementation details. This language can also be used to uniquely specify and document the logical design of data structures during the preliminary and detailed design phases of a project.

B. THE DATA STRUCTURE DEFINITION LANGUAGE

Data structures are closely related to sets; therefore, we will use concepts and notations adapted from set theory for describing the logical aspects of data structures. A data structure contains members, in the

same way that a set has members (we are excluding the null set in this discussion since a data structure with no members, and hence no information, is not of practical value). We will assign unique names to all data structures. We may assign arbitrary names to the data structure members, provided we do not use any of the data structure names. Thus, for example, we can write:

A = (A1, A2, A3, A4)
B = (B1, B2, B3, B4, B5)
C = (DOG)

Here we have defined three data structures, A, B, and C, with members as shown.

We impose the restriction that a data structure member may not be a data structure. This means that the following definition is illegal:

A = (B, C)
B = (D, E)

This restriction is imposed by the current capability of the MIDAS Translator and is expected to be removed at a later date.

We also note that the logical definition of a data structure does not require the members to be ordered. For example, the following two definitions of data structure A are logically equivalent:

A = (B, C)
A = (C, B)

However, physical implementation of data structures using the MIDAS Translator will require that the members be ordered. Therefore, we will assume that the members are ordered for this reason. The order is that in which the members are listed in the data structure definition. Since there is only one definition for each data structure, there is no possibility of a conflict in order arising.

A member may have an attribute associated with it. The only attribute which will be permitted under the initial DSDL will be the pointer attribute. The pointer attribute specifies that the member is a pointer to a specified data structure. A member with the pointer attribute is denoted

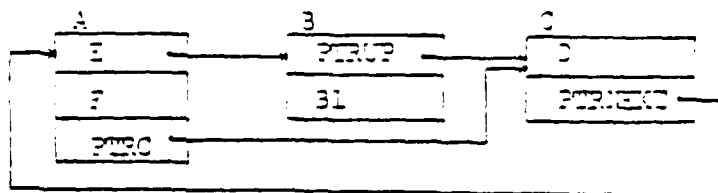
by the form $A=*B$, where A is the member name and B is the data structure which is pointed to by A. For example:

A = (E=*B, F, PTRC=*C)

B = (PTRUP=*C, B1)

C = (D, PTRNEXT=*A)

These definitions are equivalent to the following schematic representation:



Members which are data structure pointers always point to the data structure itself, not to any members of the data structure. This distinction is important to understand, since the entire MIDAS system is based on this convention. Thus, in the above example, member PTRC of data structure A points to data structure C, but not to member D of data structure C.

The above discussion treats only the logical aspects of data structure definition. In order for MIDAS to implement a data structure it is also necessary to specify the physical properties of the data structure. This is done for each member of the structure.

Each structure will be implemented in one or more words of storage. A member may occupy an entire word, or it may occupy a bit field within a word. In addition, the data structure itself is embedded within a FORTRAN structure such as an array or COMMON block. A means must be provided for specifying all of these physical properties.

DSDL will utilize a parenthesized notation to specify the physical properties of data structures. The FORTRAN structure which contains a logical data structure will be denoted as in the following example:

A(ISPACE) = (B, C, D)

This indicates that data structure A is physically contained within an array named ISPACE. Only onedimensional arrays may be used for FORTRAN structures with the current version of the MIDAS Translator.

Members of data structures are stored as one member per word, in the same order as the members are listed in the data structure definition, unless otherwise indicated. Thus, in the above example, data structure A requires three words of storage space, one each for B, C, and D.

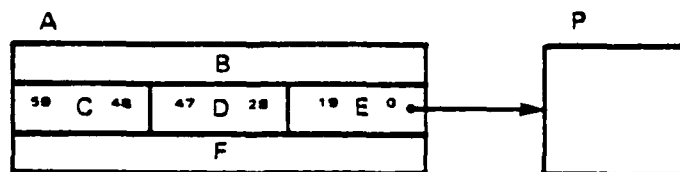
It may be desirable to pack several members into the same word, with a specific bit field allocated for each member within the word. All of the members which occupy the same physical word are enclosed by parentheses. For example:

A(ISPACE) = (B, (C, D, E), F)

Here data structure A requires three words of storage, but members C, D, and E are packed into the second word. We have shown how two or more members are packed, but we have not indicated the bit fields which contain them. Bit field designation is accomplished with an integer which specifies the number of consecutive bits in the field. This integer is enclosed in parentheses and immediately follows the member name (if the member name has an attribute, the bit field information follows the attribute information). For a given computer word, the bit fields are assigned in the order that members are specified, starting with the leftmost bit and proceeding to the right. It is not necessary that the bit space of the entire word be allocated. For example:

A(ISPACE) = (B, (C(12), D(28), E*(P(20))), F)

Assuming a 60 bit word, this data structure definition can be represented schematically as:



The Data Structure A is contained in array ISPACE. Data structure P, which is pointed to by member E of A may or may not be in ISPACE.

All data structure members which are bit packed are assumed to be accessed (in the FORTRAN sense) in the array name specified on the left hand side of the data structure definition statement. Normally, this

"default" array will be of integer type, since it is unlikely that a packed word will contain floating point data. On the other hand, other members of the same data structure which occupy entire words may be of a different number type, such as real, and therefore require access using a different FORTRAN array name.

The default array name can be overridden for a particular member by following the member name with the array name required by that member; this member array name is enclosed by parentheses and applies only to that specific member. This feature is applicable only to members which occupy entire words; all members which are bit packed automatically use the default array name. For example:

```
A(ISPACE) = (B(SPACE), (C(12), D(28), E=*P(20)), F)
```

C. TABLE GENERATION FOR THE MIDAS TRANSLATOR

Before the MIDAS Translator can be used to translate source programs written in the MIDAS language, it must be told how to interpret logical data structure references and implement them in FORTRAN. This is done by an input language which includes statements written in the Data Structure Definition Language along with other information. Using this input, the MIDAS Translator builds the translation tables from scratch, or, alternatively, augments a table which already exists.

Table construction or augmentation is known as Phase I operation of the MIDAS Translator. This phase is optional in a given MIDAS run, but must have been done at least once before midas source language programs can be translated.

Phase I operation is initiated by the appearance of a special input card which has one of two forms. If the translation tables are being constructed from scratch the initial card has the form:

```
/MIDAS dialect
```

Where dialect is one of the key words CDC, IBM, MULTICS, or UNIVAC.

Thus the dialect specifies a particular computer system for which the tables will be constructed. On the other hand, if additional data structure definitions are being added to previously generated tables, MIDAS already knows which computer system is required, and the initial card has the form:

```
/MIDAS
```

The MIDAS card is followed by information which defines the number type for all arrays which are being introduced for the first time in subsequent data structure definition statements; if an array has previously been introduced in an earlier Phase 1 run, MIDAS already knows its number type and respecification of number type for that array will not be allowed. The specification of array type is needed when tables are being built for either IBM or MULTICS dialect. For CDC and UNIVAC dialect the array type information is superfluous and need not be included.

The array type information is introduced with the card:

```
/TYPE
```

This is followed by one or more cards of array type specification; each card begins with the keyword REAL or INTEGER, followed by the names of the arrays of that type, using comma separators between array names. For example:

```
/TYPE  
REAL SPACE, SCOREBOARD, QUEUE  
INTEGER ISPACE, ISCORE  
REAL QXR
```

The data structure definition block appears next. It is introduced by the card:

/DEFINE

which is followed by one or more cards containing data structure definitions, using the Data Structure Definition Language described in the previous section.

Following the data structure definition block the user may include an optional macro definition block. This block is introduced by the card:

/MACRO

which is followed by one or more macro definitions. Each macro definition is introduced by a card containing the macro name enclosed by \$ characters. This is then followed by one or more card images which contain the actual macro text. The macro text may be arbitrary except that the first character on any macro text card can not be a \$ or /. For example:

```
/MACRO
$$UES
THIS IS A MACRO
$MARY$
MARY HAD
A
LITTLE
    MACRO
```

In this example two macros, \$\$UES and \$MARY\$, are defined.

The end of the input for Phase 1 operation is the card:

/END

Note that the only required input cards for Phase 1 operation are /MIDAS and /END. All other input is optional, however, at least one definition block must be included, as introduced by /TYPE, /DEFINE, or /MACRO.

If a previously generated table is being augmented, only additions are permitted to the table. No previously defined information may be changed or redefined.

D. THE MIDAS LANGUAGE

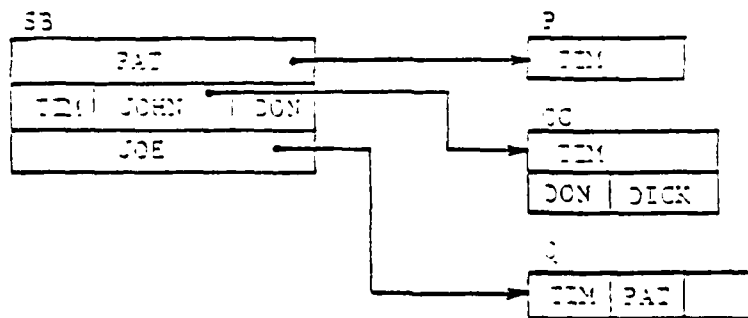
In order to illustrate the MIDAS language we define a set of data structures which will be used in conjunction with an extended example. The definitions are written using the input language for Phase 1 operation of the MIDAS Translator:

```

/MIDAS CDC
/DEFINE
SB(ABC) = (PAT=*P, (TIM(30), JOHN=*CC(10), DON(20)), JOE=*Q)
CC(XYZ) = (TIM, (DON(30), DICK(30)))
Q(AAA) = ((TIM(20), PAT(30))
P(BB) = (TIM)
/MACRO
SSUES
    INTEGER ABC
    DATA IDEBUG/0/
/END

```

These data structures have the following schematic representation:



Next we present a MIDAS subroutine which exhibits all of the features of the MIDAS language:

```
SUBROUTINE PDQ (B,C,JIM,BILL,BOB)           1
COMMON/ARRAYS/ABC(1000),XYZ9500),AAA(200),BBB(700)  2
DECLARE JIM=SB, K                               3
DECLARE J, BILL = CC                             4
DECLARE BOB = SB                                 5
$$UES$                                           6
$J=JIMS$                                         7
$K=BOB.PATS$                                     8
100 A=B*C+J*K                                   9
    B=$J.JOE.PATS$                              10
200 $K.TIMS=3+$K.TIMS*5/$BILL.DONS$            11
    RETURN                                       12
END                                              13
```

This subroutine is not intended to represent meaningful computation but only to serve as an example of the use of the MIDAS language. The lines of the subroutine are numbered to permit easy referencing in the discussion which follows.

The arrays which contain the data structure SB, CC, Q, and P must be available within the subroutine since they will be referenced by the FORTRAN subroutine generated by the MIDAS Translator. This is accomplished through the COMMON statement in line 2. The arrays could also have been established through the parameter list or with DIMENSION statements.

The data structures SB, CC, P, and Q are actually prototypes or templates. In practice, there will probably be many copies of a given type of data structure in use. The individual copies are distinguished from each other by the use of different pointers for each copy. Thus, we need a means of associating a particular pointer name with the type of data structure which it is pointing to. This is done with the DECLARE statement.

In line 3 we declare the FORTRAN variable JIM to be a pointer to a data structure of type SB. Similarly, in line 5 we declare BOB to also be a pointer to a data structure of type SB. JIM and BOB may point to the same instance of SB or to different instances. In line 4 we declare BILL to be a pointer to a data structure of type CC. It is necessary to establish values for JIM, BOB, and BILL before each of them is first used in a MIDAS data structure reference. In this example, these values are passed through the parameter list.

Given the name of a pointer to a data structure, we can now locate and reference the value of any member within that data structure. This is done by constructing a compound name, enclosed in \$ symbols, using the (.) period symbol to separate elements of the compound name. For example, we can reference the member JOHN in data structure SB, using JIM as a pointer to SB, by the notation \$JIM.JOHN\$. This provides us with the actual value of JOHN. Since JOHN is a pointer to a data structure of type CC, we can then refer to member DICK of CC by the notation \$JIM;JOHN;DICK\$.

It is useful to have a means of working with partial compound names by establishing intermediate pointer values. This avoids having to unravel a long compound name each time the name is used, with the corresponding space and time penalties in the executing program. This can be accomplished using a pointer macro. A pointer macro establishes a MIDAS name for a pointer value which has been derived by traversing a pointer chain through the data structures. Once defined, the pointer macro can then be used in place of the compound name which it represents.

Examples of the specification of pointer macros are shown in lines 7 and 8. As can be seen, the pointer macro definition consists of a simple replacement statement enclosed by \$ symbols. The left side of the replacement statement is an integer FORTRAN variable; the right side is a simple or compound name representing a data structure pointer. On the right side the simple name or the first element of the compound name may be either a data structure pointer or a previously defined pointer macro.

Once defined, a pointer macro may not be redefined. The definition remains to the end of the program or subprogram. Also, a pointer macro

must be defined before it is used. This requires that the definition card physically must precede any cards which use the pointer macro in the MIDAS source deck, and that the pointer macro definition statement must be executed before any statements which reference the pointer macro in the actual program execution.

In line 8 we define the pointer macro K as representing the string BOB.PAT; this will actually generate a FORTRAN variable K which contains the value of \$BOB.PATS, so that K may subsequently be used in the normal FORTRAN sense. However, since K also represents the string BOB.PAT, we note that the reference \$K.TIM\$ is identical to the reference \$BOB.PAT.TIM\$, as shown in line 11.

Similarly, we are substituting the MIDAS name J for JIM (and consequently the value of FORTRAN variable J is set to the value of FORTRAN variable JIM) as shown in line 7. This permits us to write the MIDAS name \$J.JOE.PATS in line 10, which is equivalent to \$JIM.JOE.PATS.

Whenever a name is to be used as a pointer macro, this fact must be noted in a DECLARE statement. This is done by simply including the name in the list for a DECLARE statement such as is done in lines 3 and 4 for K and J, respectively. Individual items in a DECLARE statement list are separated by commas as shown.

MIDAS also permits the user to employ card macro definitions. A reference to a card macro is shown in line 6 and consists of the macro name (e.g., SUE) enclosed in \$ symbols. The macro reference must not start in columns 1 through 6. Whenever a card macro is encountered, MIDAS will replace the macro with a set of one or more 80 column card images which correspond to the macro name. These card images may not contain any non-FORTRAN (i.e., MIDAS) text, since the card images are not interpreted by the MIDAS translator. The END card must not be included within the macro card set. The card macro may occur anywhere before the END card, there may be as many different card macros as desired, and the same card macro may appear several times.

There are some restrictions on ordering of MIDAS information. As discussed earlier, pointer macro definitions must precede any use of that

pointer macro both in card sequence and execution sequence. Also, the DECLARE statement must precede the use of any item specified in that DECLARE statement.

Whenever an END card is encountered, signifying the end of a subprogram, all definitions local to that subprogram as expressed on DECLARE cards and through pointer macro definition statements are lost. Other subprograms may follow and will be processed in sequence, but they must establish their own local definitions.

A word about number types: Names chosen for pointers to data structures and macro pointers (as specified in DECLARE statements) will have an identical FORTRAN name. Since these names represent pointers, they would be chosen so that they start with letters I through N. Otherwise, it will be necessary to type them as INTEGER using a TYPE statement.

E. USING THE MIDAS TRANSLATOR

The MIDAS Translator is a cross-translator, i.e., it is capable of generating output for several different computing systems including the system it operates on. Furthermore, MIDAS is designed to operate on all of the computing systems for which it is capable of providing translation output. These systems are CDC 6000/7000/Cyber series with FTN FORTRAN, IBM 360/370 series with FORTRAN Level G and H, Univac 1108 with FORTRAN V, and Honeywell MULTICS with FORTRAN.

The operating details depend in part upon the computing system on which the MIDAS Translator is operating. However, certain aspects of operation are common to all systems and will be discussed first.

1. General Operating Details

MIDAS Translator operation involves two phases. Phase 1 generates the MIDAS translation tables using the input language described in Section C. For a particular programming project and target computer a Phase 1 operation must be executed at least once. The tables which are generated may be saved as files and used for immediate or later translation of programs written in the MIDAS language. Furthermore, the tables may be augmented at any later time by additional Phase 1 operations.

Whenever a Phase I operation occurs, the tables which are generated as output are called New MIDAS Tables. Whenever the MIDAS Translator is run for strictly translation purposes or for augmenting existing MIDAS tables, an existing set of tables must be provided as input; these existing tables are called Old MIDAS Tables. Thus, every MIDAS Translator run except the initial run which builds the first set of tables for a particular project and/or target computer will require a set of Old MIDAS Tables.

The old and new tables are input and output as binary files. The file names are dependent upon the computer system on which MIDAS is running. In addition, there are three other files: card image input, printed output, and translated output.

The card image input file contains two optional blocks of information, of which at least one of the blocks must be present. The first block is the data structure definition using the language described in Section C. The other block contains the programs to be translated, written in MIDAS language. Each program to be translated must terminate with a normal FORTRAN END card, and as many programs as desired may sequentially appear in the input. If both data structure definitions and source programs appear, the data structure definitions must come first.

Listed output from the MIDAS Translator includes all input information and diagnostics in the event of error. If a source program contains a MIDAS detectable error, an error statement will be inserted into the translated output for that program which will produce a FORTRAN error when the program is compiled. Subsequent source programs will be translated correctly.

The translated output file contains card images of the translated source program in the FORTRAN dialect appropriate to the target computing system.

2. CDC Operating Details

The following file names are used by MIDAS:

INPUT	Input card images
OUTPUT	Listed output
OLDMT	Old MIDAS Tables
NEWMT	New MIDAS Tables
FTNOUT	Translated source programs

The program operates in a field length of 55000₈ words. MIDAS can be called by the single control card:

MIDAS.

If the input card images are on a file other than INPUT, say, file SOURCE, then the control card should read:

MIDAS,SOURCE.

The MIDAS Translator on CDC systems provides complete translation capability to CDC, IBM, UNIVAC, and MULTICS target systems.

3. MULTICS Operating Details

The following file names are used by MIDAS:

FILE04	Input card images
FILE06	Listed output
FILE07	Translated source programs
FILE08	Old MIDAS Tables
FILE09	New MIDAS Tables

The MIDAS Translator on MULTICS provides complete translation capability to IBM, UNIVAC, and MULTICS target systems. Data structure definitions involving bit packed words are not permitted for translation to CDC target systems.

4. IBM Operating Details

The IBM version of MIDAS is under development at the present time.

5. UNIVAC Operating Details

The UNIVAC version of MIDAS is under development at the present time.

11 May 1978

TO: MIDAS Users
FROM: A. F. Malmberg, Chief Scientist (5200)
SUBJECT: Multiple Definition Capability in MIDAS

MIDAS has been extended to permit the user to make multiple definitions of data structures and data structure members. These definitions may be made at the data structure, word, or byte level. This feature is supported by UPDATE level AM78131A and above of MIDAS.

The + symbol is used to separate the multiple definitions at a particular level. For example, two definitions of the data structure S can be made as follows:

$$S(AAA) = (J, K) + (L, M, N)$$

Note that each definition of the data structure may have a different number of words.

Multiple definitions at the word level can be accomplished as shown in this example:

$$S(AAA) = (J + P, K + B + C)$$

Multiple definitions at the byte level may also be made:

$$S(AAA) = (J, (D(3) + E, F(10), G(30) + H + W))$$

When making multiple definitions for a particular byte, the first definition listed must specify the byte size in bits; subsequent definitions of that byte must not specify the byte size.

Each data structure definition statement must contain unique names. Thus, the following multiple definition is not allowed:

$$S(AAA) = (J, K) + (L, J)$$

BDM/W-AFM-2390-78
Page 2 of 2
11 May 1978

Other than common byte size for a particular byte, there are no restrictions imposed on the nature of the members in multiple definition. Thus, each member may be independently assigned pointer attributes and alternate FORTRAN array names. For example:

$$S(AAA) = (J)+(K,(F=*B(3)+C,D(5))+M=*S(R))$$

Data structure definition statements may be continued from one card image to the next if the last non-blank character on the card to be continued is either , or +.

AFM/mlh

APPENDIX C

THE PROGRAM DESIGN LANGUAGE

A. INTRODUCTION

The algorithms and program segments for MADEM have been developed using a program design language (PDL). The use of PDL permits design to be expressed without the necessity of using an implementation in a standard programming language to specify the design detail.

Traditionally, narrative descriptions, decision tables, and flowcharts have been used to describe the design of a software system. These techniques are now being challenged by program design languages such as that used for the MADEM design. These PDLs provide: (1) a vehicle to translate functional modeling concepts into program design; (2) a replacement for design logic flowcharts; and (3) a means for communication between technical and nontechnical personnel, designers and developers.

PDL also has the advantage of having a closer relationship to programming languages than traditional methods of expressing design, thereby permitting a more direct mapping of design specification into code.

PDL is English-like in its means of expression and follows certain semantic and syntactic conventions. The concepts of structured programming are applied in the form of basic control structures for logic flow and indentation. Top down programming is implemented by specifying in PDL the top level portion of the program and evolving the PDL into succeeding levels of detail. There is considerable latitude in the selection of predicate and function descriptions which may be in English, in a computer language, or some combination of both.

The PDL is used for both the actual design framework of the program as well as for algorithms appearing only in hard copy form. The use of PDL eliminates the need for all flow charts and provides a self-documenting capability for the program itself. Realization of an implementation consists

simply of adding the necessary coding to the logical design statements. Thus, the design language and the implementation language coexist in the final source code.

B. CONTROL STRUCTURES

Four types of control structures are used: sequential composition, DO WHILE, IF-THEN-ELSE, and CASE. By suitable combination of these four types, programs and algorithms of any complexity may be expressed. These basic structures are combined with an indentation notation which is used to delineate the bounds of each structure in the program or algorithm.

The keywords, DO WHILE, IF-THEN-ELSE, CASE, etc., are written as structured comments. This permits the predicates for DO WHILE, IF-THEN, and CASE to be written in more natural terms since they do not have to be intelligible to the compiler. In implementation they are followed immediately by the associated programming language code statements. Thus, the programmer has the ability to write programs in true structured style using language elements of his own choosing, along with the implementation of the structure using a standard programming language. In this way, the coding and design documentation are carried along in one-to-one correspondence.

In the structured comment format, each comment is begun with an asterisk to visually delineate comment lines from coding lines. A comment may be continued on the next line. However, the continuation line does not have the asterisk header and starts one column farther to the right than the initial line of comment. Thus, if, for a given indentation level, the comment asterisk appears in column n , the first character of the comment continuation will start in column $n+1$. All implementation statements also start in column $n+1$; if an implementation statement is continued, all such continuation lines start at column $n+2$. Thus, the structured skeleton of the program design in PDL is easily followed by the identifying asterisks which introduce the structured comments. Each subsequent indentation level is started three columns farther to the right from the previous level to indicate nested structures.

Sequential compositions are shown diagrammatically in Figure C-1 and are written on the same indentation level, as in this example:

```
*INITIALIZE CONTROL VARIABLES
*READ INPUT DATA
*BUILD DATA STRUCTURE
*TRAVERSE DATA STRUCTURE, GENERATING OUTPUT
```

The IF-THEN-ELSE structure shown in Figure C-2 is illustrated by:

```
*IF(predicate)THEN
  *BLOCK 1
*ELSE
  *BLOCK 2
*BLOCK 3
```

If the predicate is satisfied, then BLOCK 1 is executed. BLOCK 1 is automatically terminated by the appearance of ELSE on the same level as the IF-THEN. Thus, following completion of BLOCK 1, control is passed to BLOCK 3 (which is actually a sequential block following the IF-THEN-ELSE block, appearing here only for purposes of illustration). If the predicate is not satisfied, then BLOCK 2 is executed, followed by BLOCK 3.

An alternate form of the IF-THEN-ELSE structure is shown in Figure C-3 and occurs when there is no ELSE part:

```
*IF(predicate)THEN
  *BLOCK 1
*BLOCK 3
```

If the predicate is satisfied, then BLOCK 1 is executed, followed by BLOCK 3. If the predicate is not satisfied, BLOCK 1 is not executed, and control passes immediately to the execution of BLOCK 3.

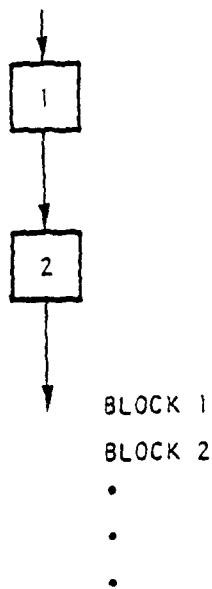


Figure C-1. Sequential Composition

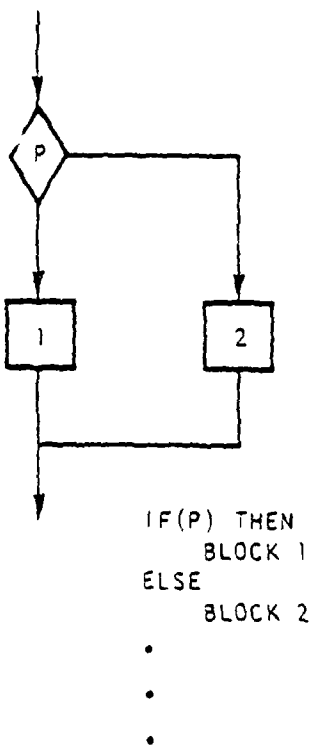
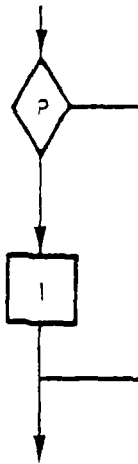


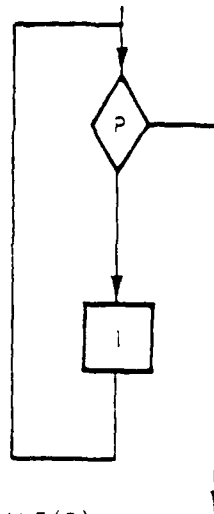
Figure C-2. If-Then-Else Structure



IF(P) THEN
BLOCK I

•
•
•

Figure C-3. If-Then Structure



DO WHILE(P)
BLOCK I

•
•
•

Figure C-4. Do While Structure

The DO WHILE structure is shown in Figure C-4 and has the form:

```
*DO WHILE(predicate)
  *BLOCK 1
*ENDDO
*BLOCK 2
```

BLOCK 1 is executed until the predicate becomes FALSE. The use of the ENDDO statement is optional.

A variant on the DO WHILE is the DO, sometimes a more natural means of expressing the conditions for executing the subordinate block. Its action is identical to the DO WHILE:

```
*DO(predicate)
  *BLOCK 1
*BLOCK 2
```

Another type of control structure is the CASE structure shown in Figure C-5. This is used to select one of several possible blocks, depending upon the value of an expression. Its structure can be seen by the following example:

```
*CASE(expression)
  *Expression value = 1
    *BLOCK 1
  *Expression value = 2
    *BLOCK 2
  .
  .
  .
  *Expression value = n
    *BLOCK n
*BLOCK A
```

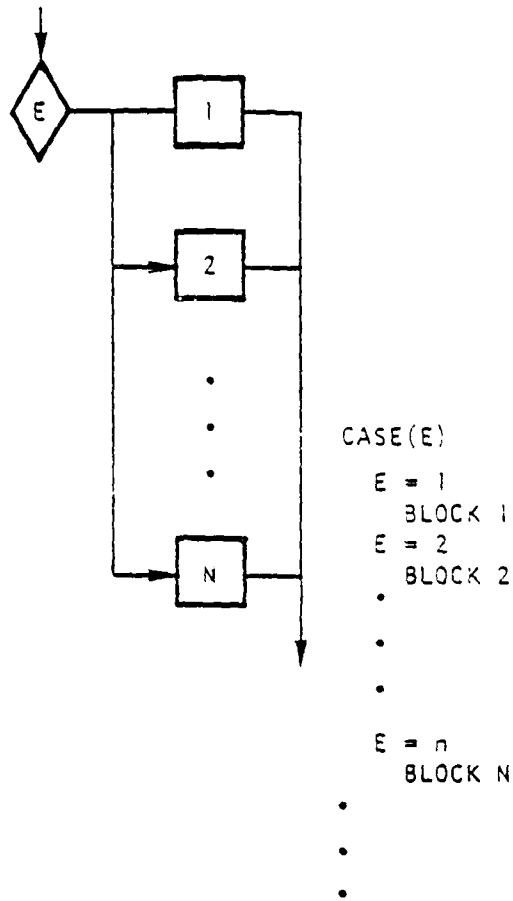


Figure C-5. Case Structure

The expression is evaluated and control is passed to a particular block, depending upon the expression value. Following completion of the specified block, control is passed out of the CASE structure to the next sequential block, BLOCK A.

C. SEGMENTATION

Programs are subdivided into segments, both for purposes of multiple references and to aid intellectual manageability. Each segment is labeled with a name to identify it. The name may be any English phrase of one or more words which serve to identify the segment, usually chosen to describe its function. The segment identification occurs as the first structure comment, using the keyword SEGMENT for identification:

```
*SEGMENT(GENERATE OUTPUT DISPLAY)
```

The segment identification may then be followed with arbitrary comments which provide further information on the purpose of the segment and its data and control interface, as needed. These explanatory comments do not use the asterisk header so that there will be no accidental confusion with the actual control structures which follow.

Segments may be referenced by using the keyword INCLUDE as in this example:

```
*INCLUDE(GENERATE OUTPUT DISPLAY)
```

The close of a segment is always indicated by the statement:
END

D. PROGRAM DESIGN AND IMPLEMENTATION

The program design process involves a complete logical specification of the program in PDL. All logical steps are expressed in the language

so that another person can fully comprehend the logical process being specified.

Combining these ideas permits the writing of algorithms and programs of any complexity. As an example, consider the following segment which specifies an algorithm for traversing an n-ary tree, where each node of the tree is represented by a linked list of sibling cells, each of which has a pointer to a descendant node. Thus, each sibling cell has two pointers, one for the sibling cell and one for the descendant node.

```
*SEGMENT(TRAVERSE N-ARY TREE)
*ENTER ROOT NODE
*TRAVERSE=TRUE
*BACKUP=FALSE
*DO WHILE(TRAVERSE)
  *IF(NODE CELL HAS DESCENDANT AND BACKUP=FALSE)THEN
    *DESCEND TO DESCENDANT CELL
  *ELSE
    *BACKUP TO PARENT CELL
    *IF(ROOT NODE)THEN
      *TRAVERSE=FALSE
    *BACKUP=TRUE
END
```

When the PDL structure is implemented, the programming language statements are inserted immediately following the corresponding PDL statement. In many cases, the implementation of a simple PDL structure will contain additional microstructure. Such microstructure should also follow the same rules for indentation of its logical elements.

As an example of the implementation of PDL into FORTRAN, consider the implementation of the above example for traversal of n-ary trees. The tree cells are stored in a FORTRAN array IQ with a pointer LTREE for the root cell of the tree. Each cell contains two pointers, with the first

cell word containing the sibling pointer and the second cell word containing the descendant pointer. A pointer value of zero indicates a null pointer which terminates the pointer chain. The implemented segment can thus be written as:

```

C      *SEGMENT(TRAVERSE N-ARY TREE)
      SUBROUTINE TRAVERSE(LTREE)
C      A PUSHDOWN STACK ISTACK WITH POINTER J IS USED TO BACKTRACK
C      TOWARDS ROOT NODE OF TREE
      COMMON/TREE/IQ(1000)
      DIMENSION ISTACK(100)
C      *ENTER ROOT NODE
      L=LTREE
      J=1
C      *TRAVERSE=TRUE
      ITRAV=1
C      *BACKUP=FALSE
      IBACK=0
C      *DO WHILE(TRAVERSE)
1000   IF(ITRAV.EQ.0) GO TO 2000
C      *IF(NODE CELL HAS DESCENDANT AND BACKUP=FALSE)THEN
      IF(IQ(L+1).EQ.0.OR.IBACK.NE.0) GO TO 1100
C      *DESCEND TO DESCENDANT CELL
      ISTACK(J)=L
      J=J+1
      L=IQ(L+1)
      GO TO 1500
C      *ELSE
C      *IF(NODE CELL HAS SIBLING CELL)THEN
1100   IF(IQ(L).EQ.0) GO TO 1200
C      *MOVE TO SIBLING CELL
      L=IQ(L)
C      *BACKUP=FALSE
      IBACK=0
      GO TO 1500
C      *ELSE
C      *BACKUP TO PARENT CELL
1200   J=J-1
      L=ISTACK(J)
C      *IF(ROOT NODE)THEN
      IF(L.NE.LTREE) GO TO 1300
C      *TRAVERSE=FALSE
      ITRAV=0
C      *BACKUP=TRUE
      IBACK=1
1300
1500   GO TO 1000
2000   RETURN
      END

```

APPENDIX D

MIDAS TABLES

```

/MIDAS CDC
/DEFINE
ABINFO( ISPACE) = (( NEXT == ABINFO( 30), NRACTYP( 30)),
  ( NOACOH( 30), PTRACDB == ACDB( 30)),
  NORMRQ,
  NOREARMQ,
  NOREFUELR,
  NOLAUNCHQ,
  NOUSE )
ABQUEDB( ISPACE) = (( PNEXT == ABQUEDB( 30), CLASS( 30)),
  VALUE1( SPACE),
  VALUE2( SPACE),
  VALUE3( SPACE))
ABSTATUS( ISPACE) = (( PACTAB == ABINFO( 30), NOACTAB( 30)),
  ( PTR2QUES == QUEUES( 30), NOACONAB( 30)),
  ABDAMAGE( SPACE))
ABVCR( ISPACE) = (( PNEXT == ABVCR( 30), PTRABSB == SB( 30)),
  ( PACTAB == ACTAB( 30), NOACTAB( 30)))
ACDB( ISPACE) = (( NEXT == ACDB( 30), NRACTYPE( 30)),
  MAXSPEED( SPACE),
  CRUISESPEED( SPACE),
  MAXALTITUDE( SPACE),
  MINALTITUDE( SPACE),
  MAXCLIMBIVE( SPACE),
  FUELCONSUME( SPACE),
  ACORANGE( SPACE),
  RADARCS( SPACE),
  ATTACKRADIUS( SPACE),
  MAXFUEL( SPACE))
ACDBUF( ISPACE) = (( PTRACD == ACDEVICE( 30), NUMDEV( 30))
ACDEVICE( ISPACE) = (( PNEXT == ACDEVICE( 30), TYPE( 30)),
  ( WORKING( 30), PACQDB == ACDB( 30)),
  ( JAM( 30), LEVEL( 30)))
ACRFTLIST( ISPACE) = (( PNEXT == ACRFTLIST( 30), ACRFTID( 30)),
  NUMACRFT,
  FORMTYPE)
ACRFTONAB( ISPACE) = (( PNEXT == ACRFTONAB( 30), ABID( 30)),
  ( ACRFTLIST( 30), NUMELONS( 30))
ACTAB( ISPACE) = (( PNEXT == ACTAB( 30), NRACTYP( 30)),
  ( NOACOH( 30), NOACASN( 30)))
ADLINK( ISPACE) = (( NEXT == ADLINK( 30), ID( 30))
ADSITEDB( ISPACE) = (( PNEXT == ADSITEDB( 30), ADTYPE( 30)),
  ( MODVAL1( 30) + INSTANT, MAXDIGEST( 30)),
  ( MAXCYCLEDIGEST( 30), MINTIMEDIGEST( 30)),
  LOSTTIME( SPACE),
  LASTCHANCE( SPACE) + LOWRESPTIME( SPACE),
  ENGAGEWINDOW( SPACE),
  MODVAL2( SPACE),
  MODVAL3( SPACE),
  ( COMMONONE( 30), ONE( 30)),
  ( COMMONFEW( 30), FEW( 30)),

```

```

(COONMANY(30), MANY(30)),
TIMEFLIGHT(SPACE),
MISSILERANGE(SPACE),
(MAXASSIGN(30), MODVAL4(30)+DILSIZE),
MODVAL5(SPACE)+TSEENTAC(SPACE),
MAXTRACKRANGE(SPACE)+TCHKCOV(SPACE),
LOCKONTIME(SPACE)+TSEEKENG(SPACE),
MODVAL6(SPACE)+HILIM(SPACE),
MODVAL7(SPACE)+LOWLIM(SPACE),
CONVLOAD,
(SNUKLOAD(30), LNUKLOAD(30)),
(CVRESUPPLYFREQ(30), RESUPPLYCV(30)),
(SNRESUPPLYFREQ(30), RESUPPLYSN(30)),
(LNRESUPPLYFREQ(30), RESUPPLYLN(30))
ALLOCATE(1SPACE)=( (FUP=>ALLOCATE(30), PDOWN=>ALLOCATE(30)),
(AMMOTYPE(30), PFU=>FIRELINIT(30))
ANYDIL(1SPACE)=( (FUP=>ANYDIL(30), PDOWN=>ANYDIL(30)),
(PPERC=>PERLIST(30), PDIB=>DIB(30)),
(LOC(30), PRIORITY(30)),
(PRIUP=>ANYDIL(30), PRIDN=>ANYDIL(30)),
(DCOV(30), SHORT(30)),
(PDELAY=>DAGE(30), PENGEV=>EVENT(30)),
(PAL(30), NULL(30))
AGDB(1SPACE)=( (NEXT=>AGDB(30), NRACTYP(30)),
RANGE(SPACE),
NOUSE1,
NOUSE2)
ARCFTSAW(1SPACE)=( (FNEXT=>ARCFTSAW(30), FSB=>SB(30)),
(ADDRESS=>HEX(30), TYPE(30)),
DAMAGE(SPACE))
ARCFTSTATUS(1SPACE)=( (PFLTDB=>FLTDB(30), FMUNITIONS=>MUN(30)),
(PSTRTH=>HEX(30), PENDHX=>HEX(30)),
(PNXTHX=>HEX(30), PAIRBASE=>SB(30)),
(NULL1(30), PAIRTGT=>SB(30)),
(PCNDTGT=>SB(30), NUMAIRCRAFT(30)),
(FLITELEG(4), INTERCEPTSTATUS(4), ALTUDECHNG(4),
PROFILENDX(4), LANDING(4), ORBITSTAT(4), AIRCOMBAT(4),
NULL2(4), JAMSTAT(4), DUMMY(24)),
FUEL,
ALTITUDE(SPACE),
SPEED(SPACE),
DIRECTION(SPACE))
ATTACKBLOK(1SPACE)=( (FNEXT=>ATTACKBLOK(30), FNXTGTL=>STDBLOK(30)),
(PFMKTG=>FAKTGBLOK(30), NOFMKT(30)),
(ISECTOR(30), FNXTCRD=>CORRIDORBLOK(30)))
BLOCHEAD(1SPACE)=( (STATUS(24), TIMEBOT(17), ALLOCATOR(10), TYPE(3))
BOCDIL(1SPACE)=( (FUP=>BOCDIL(30), PDOWN=>BOCDIL(30)),
(PPERC=>PERLIST(30), PDIB=>DIB(30)),
(LOC(30), PRIORITY(30)),
(PRIUP=>BOCDIL(30), PRIDN=>BOCDIL(30)),
(DCOV(30), SHORT(30)),
(PDELAY=>DAGE(30), PENGEV=>EVENT(30)),
(PPAL=>PAL(30), PSUB=>SUBLIST(30))
BOCNOCHANMSG(1SPACE)=(NULL1,
NULL2,
..

```

```

NULL3,
(NULL4(30), PTGTSB=>SB(30))
BOCSTAT( ISPACE) = (( AUTO(30), READY(30)),
(PSDIG=>EVENT(30), PSDELY=>EVENT(30)),
(PADIL=>BOCDIL(30), NADIL(30)),
(PHFOQ=>BOCDIL(30), PTFQ=>BOCDIL(30)),
(PHNDIL=>PERLIST(30), PTNDIL=>PERLIST(30)),
(HPRIOR=>BOCDIL(30), TPRIOR=>BOCDIL(30)),
(PDIL=>BOCDIL(30), NUMRDY(30)),
(PHDAG=>DAGE(30), PTDAG=>DAGE(30)))
BOGTSTLSTMSG( ISPACE) = (NULL1,
NULL2,
NULL3,
(NULL4(30), PTGTSB=>SB(30)))
BORBSTAT( ISPACE) = (( AUTO(30), READY(30)),
(PSDIG=>EVENT(30), PSDELY=>EVENT(30)),
(PADIL=>ANYDIL(30), NADIL(30)),
(PHFOQ=>ANYDIL(30), PTFQ=>ANYDIL(30)),
(PHNDIL=>ANYDIL(30), PTNDIL=>ANYDIL(30)),
(HPRIOR=>ANYDIL(30), TPRIOR=>ANYDIL(30)),
(ODIL=>ANYDIL(30), CAPACITY(30)),
(PHDAG=>DAGE(30), PTDAG=>DAGE(30)))
BTRYDIL( ISPACE) = (( PUP=>BTRYDIL(30), PDOWN=>BTRYDIL(30)),
(PPERC=>PERLIST(30), PDIB=>DIB(30)),
(LDC(30), PRIORITY(30)),
(FRIUP=>BTRYDIL(30), FRIEN=>BTRYDIL(30)),
(DCOV(30), SHORT(30)),
(PDELAY=>DAGE(30), FENGEV=>EVENT(30)),
(PAL=>ALLOCATE(30), NUKE(30)),
(CEASE(30)+PDIBASN=>DIB, ABENPRIOR(30)),
(WAITTOTRACK(30), WAITTOFIRE(30)),
START( ISPACE),
END( ISPACE))
BTRYFIREMSG( ISPACE) = (NULL1,
NULL2,
NULL3,
(MISSILES(30), EMPTY(30)))
BTRYSTAT( ISPACE) = (( AUTO(30), NUM(30)),
(PSDIG=>EVENT(30), PSDELY=>EVENT(30)),
(PADIL=>BTRYDIL(30), NADIL(30)),
(PHFOQ=>BTRYDIL(30), PTFQ=>BTRYDIL(30)),
(PHNDIL=>PERLIST(30), PTNDIL=>PERLIST(30)),
(HPRIOR=>BTRYDIL(30), TPRIOR=>BTRYDIL(30)),
(TRACKED=>BTRYDIL(30), IDLE(30)),
(PHDAG=>DAGE(30), PTDAG=>DAGE(30)),
(PTL(30), A1(30)+CAMMO),
(RESUPPLY(30), A2(30)+N1AMMO+TOTAMMO),
(NUCNO(30), A3(30)+N2AMMO),
(AMTOT(30), NUMEND(30)))
BYCEASEMSG( ISPACE) = (NULL1,
NULL2,
NULL3,
(PTGTSB=>SB(30), NULL4(30)))
BYENGPROGMSG( ISPACE) = (NULL1,
NULL2,

```

```

NULL3,
(PGTGSB==*SB(30), NULL4(30))
BYFLTATTMSG( ISPACE)=(NULL1,
NULL2,
NULL3,
(PGTGSB==*SB(30), NKILLED(30)),
TIME( SPACE ))
BYNOCHANMSG( ISPACE)=(NULL1,
NULL2,
NULL3,
(PGTGSB==*SB(30), NULL4(30)),
TIME( SPACE ))
BYRELOADMSG( ISPACE)=(NULL1,
NULL2,
NULL3,
(ROUNDS(30), NRFP(30)))
BYTRKCHGMSG( ISPACE)=(NULL1,
NULL2,
NULL3,
(PGTGSB==*SB(30), NULL4(30)),
TIME( SPACE ))
BUFFER( ISPACE)=( (PSTART==*LINK(30), NUMLINK(30))
C2( ISPACE)=(UNITNUMBER,
(PUP==*C2(30), PDOWN==*C2(30)),
(PSB==*SB(30), FNEXT==*C2(30)),
(UNITTYPE(30), SIDE(30)))
C2SIDE( ISPACE)=( (PTRDOWN==*C2(30), SIDE(30))
CHAINBLKR( ISPACE)=( (NULL(30), FNEXT==*CHAINBLKR(30)))
COMMAND( ISPACE)=( (FNEXT==*COMMAND(30), NUMACTS(30)),
(TMFLG(30), ADDRESS==*HEX(30)),
TIME( SPACE ) + ACTION)
CONSTBLK( ISPACE)=(CORDSLOPE( SPACE),
YINTLBUF( SPACE),
YINTLCOR( SPACE),
YINTRCOF( SPACE),
YINTRBUF( SPACE),
XSPREAD( SPACE),
YSPREAD( SPACE),
ENTRYSLOPE( SPACE),
YLINENTRY( SPACE),
YENDCORD( SPACE))
CORRIDORBLOK( ISPACE)=( (FNEXT==*CORRIDORBLOK(30), NRCORD(30)),
(PABVCOR==*AEVCR(30), NOAEVCR(30)),
(PLHEX==*HEX(30), PRHEX==*HEX(30)),
(PCHEX==*HEX(30), NRWIDTH(30)),
(PHLIST==*HEXBLOK(30), PBDNST==*CONSTBLOK(30)),
DEPTHOR( SPACE),
ANGCORD( SPACE),
ANGSPRD( SPACE),
BUFFWDH( SPACE))
COVERFU( ISPACE)=( (FNEXT==*COVERFU(30), PAL==*ALLOCATE(30)),
(NULL(30), AMMO(30)))
CRCDEADBLOK( ISPACE)=( (PSEERSB(30), NULL(30)),
(DEADUNITSB(30), UNITTYPE(30)))
CRCMSGDATA( ISPACE)=( (MSGTYP(30), FENDRSB(30)),
..

```

```

(PGTGSB(30)+PDEADSB+PMDVRSB+PLOSTSB+PINTSB.
TGTTY(30)+DEADTY+ADDRESS+NULL+NUMACFT))
CRCSEEBLUE( ISPACE)=( (PNEXT==CRCSEEBLUE(30), ID(30)),
(PSB==SB(30), ADDRESS==HEX(30)))
CRCSEERED( ISPACE)=( (PNEXT==CRCSEERED(30), ID(30)),
(PSB==SB(30), ADDRESS==HEX(30)),
(RPT(30), HUNTER(30)),
DIRECTION( SPACE),
(PNXTSER==SEER(30), NUMSEE(30)))
CRCSEES( ISPACE)=( (REDSEE==CRCSEERED(30), NUMRED(30)),
(BLUESEE==CRCSEEBLUE(30), NUMBLUE(30)))
CRCSUBORD( ISPACE)=( (PNEXT==CRCSUBORD(30), ID(30)),
(PSB==SB(30), ADDRESS==HEX(30)),
(ABSCRMBLFLG(30), BTNAENDCNT(30))+
ACRFTASNZAB+
NTRCPTRASNMNT+
NUMACONAB))
DAQE( ISPACE)=( (PPREV==DAQE(30), PNEXT==DAQE(30)),
(ACTID(30), PTR(30)),
(ARG1(30), PNDI==DAQE(30)),
(ARG2(30), ARG3(30)))
DATBLK( ISPACE)=( (PNEXT==DATBLK(30), CLASS(30)),
(PADSITE==ADSITE(30)+
FFORM==FDBDELOK+
PFLTDB==FLTDB+
PACDB==ACDB+
PPAYBUF==PAYBUF+
PPROFIL==PROF ILEDELOK+
PARDB==ARDB+
PABQUE==ABQUEDB+
PACONAB==ACRFTONAB+
PCCLASS, NUMELOK(30)))
DATBUF( ISPACE)=( (PDATELK==DATBLK(30), NUMBLOK(30))
DBCLASSBLOK( ISPACE)=( (PNEXT==DBCLASSBLOK(30), CLASS(30)),
(PTR(30), NUMBER(30)))
DEATHLOK( ISPACE)=( (PSB==SB(30), PADR==HEX(30)),
(UNITTYPE(30), SIDE(30)))
DIB( ISPACE) = (TIME( SPACE),
(SIDE(30), NUMAC(30)),
(LOST(30), POSITION(30)),
HEADING( SPACE),
VELOCITY( SPACE),
ALTITUDE( SPACE))
DUMMYBLOK( ISPACE)=( (PNEXT(30), NEWKEY(30)))
ENGDATUPMSG( ISPACE)=( NULL1,
NULL2,
NULL3,
(PGTGSB==SB(30), COVER(30)),
STARTOPP( SPACE),
ENDOPP( SPACE),
(PDIB==DIB(30), PRIORITY(30)))
ENGRSULT( ISPACE)=( (NKILLED(30), PFU(30)))
EVENT( ISPACE)=( (NEHMEN==SB(30), INCDNT(30)),
(PTRUP==EVENT(30), PTRDOWN==EVENT(30)),
(MSG(30), LASSEN==SB(30)),

```

```

TIME(SPACE))
FAKTGBLOK( ISPACE)=( (PNEXT=>FAKTGBLOK(30), PNFRDB=>FDBDBLOK(30)),
( PFLTAKT=>FLTAKTBLOK(30), NOFLAKT(30)))
FDBDBLOK( ISPACE)=( (PNEXT=>FDBDBLOK(30), NRFORM(30)),
( PTRFLT=>FMFLTDB(30), NOFLT(30)),
SPFORMC(SPACE))
FIREUNIT( ISPACE)=( (AMMO(30), PNEXT=>FIREUNIT(30)),
( PDIL=>BTRYDIL(30), STAGE(30)),
( CEASE(30), PENGEV=>EVENT(30)),
( CAMMO(30), BUSY(30)),
( NIAMMO(30), NZAMMO(30)))
FLTAKTBLOK( ISPACE)=( (PNEXT=>FLTAKTBLOK(30), FNFLDB=>FLTDB(30)),
( NOACFLT(30), PFLABS=>SB(30)),
( PNACAB=>ACTAB(30), PTRFRAG=>COMMAND(30)),
PFLTSB=>SB)
FLTDB( ISPACE)=( (FNFLDB=>FLTDB(30), NRFLITE(30)),
( PTYPLOS=>PAYLOAD(30), NOPYLOS(30)),
( PTYARDB=>ARDB(30), PTACDB=>ACDB(30)),
( MAXNOAC(30), MINNOAC(30)),
( MULTAC(30), PROFILE=>PROFILEDBLOK(30)),
SPFLTC(SPACE),
DISTSEP(SPACE))
FLTDEADMSG( ISPACE)=(NULL1,
NULL2,
NULL3,
(PSB=>SB(30), PADR=>HEX(30)),
(UNITTYPE(30), SIDE(30)))
FMFLTDB( ISPACE)=( (PNEXT=>FMFLTDB(30), PNFLDB=>FLTDB(30))
FOREST( ISPACE)=( (PNEXT=>FOREST(30), NRTYPE(30)),
PTREE=>TGPTREE)
FORMATION( ISPACE)=( (FFORM=>WINGMAN(30), NUMFLTS(30))
FORMATIONBLOK( ISPACE)=( (PNEXT=>FORMATIONBLOK(30), PNFRDB=>FDBDBLOK(30)),
( NOFRMRB(30), NOFRMAL(30)))
FORTSTBUFFER( ISPACE)=( (FFOREST=>FOREST(30)+
PTGT, VARWORD(30)))
GARBLOK( ISPACE)=( (NULL(30), PNEXT=>GARBLOK(30))
HEADBLOK( ISPACE)=( (PLIST(30), KOUNTER(30))
HEX( ISPACE)=( (HEXNUMBER(30), LEVEL(30)),
( PUP=>HEX(30), PDOWN=>HEX(30)),
( TERRAIN(30), PNEXT=>HEX(30)),
( PUOL=>BUFFER(30), PEEPER=>BUFFER(30)))
HEXBLOK( ISPACE)=( (TOTAL,
( PHEX4=>HEXLINK(30), NOLIST1(30)),
( PHEX1=>HEXLINK(30), NOLIST2(30)),
( PHEX3=>HEXLINK(30), NOLIST3(30)),
( PHEX6=>HEXLINK(30), NOLIST4(30)),
( PHEXUNK=>HEXLINK(30), NOLISTUNK(30)))
HEXELEV( ISPACE)=( ELEVAT(SPACE))
HEXLINK( ISPACE)=( (PNEXT=>HEXLINK(30), HEX(30))
LEFTREE( ISPACE)=( (TIME(SPACE),
( PEVENT=>EVENT(22), DIST(4), PLEFT=>LEFTREE(17),
PRITE=>LEFTREE(17)))
LINK( ISPACE)=( (PNEXT=>LINK(30), PSB=>SB(30))
LISTENDPTRS( ISPACE)=( (PHEAD(30), PTAIL(30)+NENT))
LOAD( ISPACE)=( (PNEXT=>LOAD(30), TYPE(30)),
..

```

```

      (AMOUNT(30), POFDDB=>PAYLOAD(30))
LOSTCOVER( ISPACE )=( (FNEXT=>LOSTCOVER(30), PDIL=>BOCDIL(30)),
      (NULL(30), PRIORITY(30)) )
LOSTSIGHTBLK( ISPACE )=( (NULL(30), NOSEER(30)),
      (ARCFT(30), ADDRESS(30)) )
MESSAGE( ISPACE )=( (PTR(30), FREQ(30)),
      (VALUE1(30), VALUE2(30)) + (VALUES(30)),
      (TYPE(30), LENGTH(30)) )
MISSILEFIRING( ISPACE )=( (AMMDTYPE(30), PFIJ(30)),
      TIMEINTERCEPT( SPACE ) )
MUN( ISPACE )=( (PAG=>LOAD(30), NUMAG(30)),
      (PAA=>LOAD(30), NUMAA(30)) )
NOAVAILBLK( ISPACE )=( (FNEXT=>NOAVAILBLK,
      PJDAMAGE( SPACE ),
      PSTDBLCK=>STDBLCK ) )
ORDERS( ISPACE )=( (PTRFORMS=>FORMATION(30), PTRACT=>COMMAND(30)) )
PAL( ISPACE )=( (PUPT=>PAL(30), PDOWNT=>PAL(30)),
      (PUPE=>PAL(30), PDOWNE=>PAL(30)),
      START( SPACE ),
      END( SPACE ),
      (COVER(30), PDIL=>BOCDIL(30)),
      (PEVENT=>EVENT(30), PSUB=>SUBLIST(30)) )
PATENGAGE( ISPACE )=( (PUP=>PATENGAGE(30), PDOWN=>PATENGAGE(30)),
      (PDIL=>BTRYDIL(30), STAGE(30)),
      (CEASE(30), FENGEV=>EVENT(30)) )
PAYBUF( ISPACE )=( (FNEXT=>PAYBUF(30), NRPOCLE(30)),
      (PAYLDB=>PAYLDBLCK(30), NUMBLCK(30)) )
PAYLDBLCK( ISPACE )=( (NEXT=>PAYLDBLCK(30), TYPEINDEX(30)) )
PAYLOAD( ISPACE )=( (PNXTYPD=>PAYLOAD(30), NRPOCLE(30)),
      (MAXAMT(30), MINAMT(30)),
      (MAXFIRERANCE(30), PAYLDB=>PAYLDBLCK(30)) )
PERLIST( ISPACE )=( (PUP=>PERLIST(30), PDOWN=>PERLIST(30)),
      (PSB=>SB(30), PDIL=>ANYDIL(30)),
      (SEEN(30), PSS=>SOURCE(30)),
      (PIJCHN=>PERLIST(30), PIONCHN=>PERLIST(30)),
      TIME( SPACE ) )
PLAYERSBUFFER( ISPACE )=( (PTRPL=>PLYLIST(30), VARWORD(30)) )
PLBUFFER( ISPACE )=( (PTRPL=>C2(30), VARWORD(30)) )
PLYLIST( ISPACE )=( (ILW(30), IRW=>PLBUFFER(30)) )
POSSCOVER( ISPACE )=( (FNEXT=>POSSCOVER(30), PDIL=>BOCDIL(30)+
      PBAT=>SUBLIST),
      (PPAL=>PAL(30), PRIORITY(30)) )
PROFILEBLK( ISPACE )=( (PNXPRDD=>PROFILEBLK(30), NRPOFL(30)),
      ALTREN( SPACE ),
      ALTOGT( SPACE ),
      ALTOAG( SPACE ) )
PLBUFFER( ISPACE )=( (PSTART=>LINK(30), NUMLINK(30)) )
QUESTAT( ISPACE )=( (REALNUMBER( SPACE ) ) )
QUEUES( ISPACE )=( (NEXT=>QUEUES(30), QUENUM(30)),
      (PTR(30), NUMBER(30)),
      (PODB=>ABQUEDB(30), PQUESTAT(30)) )
RAIDBLK( ISPACE )=( (FNEXT=>RAIDBLK(30), NRRRAID(30)),
      (PTRWAVE=>WAVEBLK(30), NOWAVES(30)),
      (PTRCORD=>CORRIDORELCK(30), NDCORDS(30)) )
READYQUE( ISPACE )=( (FNEXT=>READYQUE(30), NRACTYP(30)),

```

```

(NULL(30), NUMACRT(30))
RELLIST( ISPACE ) = ( ( PNEXT = *RELLIST( 30 ), PFU = *FIREUNIT( 30 ) ) )
SAMASSIGNMSG( ISPACE ) = ( NULL1,
    NULL2,
    NULL3,
    ( PTGTSB = *SB( 30 ), TGTIL( 30 ) + NCOV ),
    STARTOFF( SPACE ),
    ENDOFF( SPACE ),
    ( PDIB = *DIB( 30 ), TGTFRICR( 30 ) ),
    ( EMPTY( 30 ), DCDV( 30 ) ) )
SB( ISPACE ) = ( ( ADDRESS = *HEX( 30 ), FC2 = *C2( 30 ) ),
    ( PSDB = *SDB( 30 ), PFEL = *EVENT( 30 ) + CUMLTIVDAMAGE,
    ( PACQ = *ACQBUF( 30 ), ID( 30 ) ),
    ( DATABASE( 30 ) +
    ABFORMTYP,   PABSTATUS = *ABSTATUS( 30 ) +
    PARCFTSTAT = *ARCFTSTATUS +
    PBOCSTAT = *BOCSTAT +
    PBTRYSTAT = *BTRYSTAT +
    REDARCFTUPCNT +
    STATUS ) )
SUB( ISPACE ) = ( ( PSB = *SB( 30 ), PSEEBUF = *SEEBUF( 30 ) + PSEE = *CROSSSES ),
    ( SUBORDINATE = *SUB( 30 ), ORD = *ORDERS( 30 ) +
    PRAID = *RAIDBLK ) )
SEEBUF( ISPACE ) = ( ( PTRSEE = *ARCFTSAW( 30 ), NUNITS( 30 ) ) )
SEER( ISPACE ) = ( ( PNEXT = *SEER( 30 ), PSEERSB( 30 ) ) )
SOURCE( ISPACE ) = ( ( PSB = *SB( 30 ), PNEXT = *SOURCE( 30 ) ) )
STOBLK( ISPACE ) = ( ( PNEXT = *STOBLK( 30 ), PTGTSB = *SB( 30 ) ),
    PTGTLTR = *TGTPTREE,
    DAMGPER( SPACE ),
    PADRPER = *HEX )
SUB( ISPACE ) = ( ( PSUB = *SUBTYPE( 30 ), NUMBER( 30 ) ) )
SUBLIST( ISPACE ) = ( ( PSB = *SB( 30 ), PNEXT = *SUBLIST( 30 ) ),
    ( AUTO( 30 ) + DEATHMARK, NUMFU( 30 ) ),
    ( RAMMO( 30 ), LOAD( 30 ) ),
    ( PDELAY = *DAGE( 30 ), NOUSE( 30 ) ),
    ( PPAL = *PAL( 30 ), ADDRESS = *HEX( 30 ) ) )
SUBTYPE( ISPACE ) = ( ( PNEXT = *SUBTYPE( 30 ), TYPE( 30 ) ),
    ( PTRSUB = *CRCSUBORD( 30 ), NUMBER( 30 ) ) )
TARGETBLK( ISPACE ) = ( ( PNEXT = *TARGETBLK( 30 ), NRTGTYP( 30 ) ),
    ( PTRFORM = *FORMATIGNBLK( 30 ), NOFORM( 30 ) ),
    ( PTGTATK = *ATTACKBLK( 30 ), NOTGTAK( 30 ) ),
    ( MAXACAL( 30 ), NOACALC( 30 ) ),
    ( MAXRHEX( 30 ), MINRHEX( 30 ) ) )
TARGETBUFFER( ISPACE ) = ( ( PTR = *C2( 30 ), VARWORD( 30 ) ) )
TARGETLISTBLK( ISPACE ) = ( ( PTRTYTL = *TTDBLK( 30 ), NOTYTPL( 30 ) ) )
TGTPTREE( ISPACE ) = ( ( DAMAGE( SPACE ),
    ( INVALID( 1 ), PSTDBLK = *STDEBLK( 21 ), DIST( 4 ),
    PLEFT = *TGTPTREE( 17 ), PRITE = *TGTPTREE( 17 ) ) ) )
TRACKINGBLK( ISPACE ) = ( ( NULL( 30 ), PSEERSB( 30 ) ),
    ( PMOVRSB( 30 ), ADDRESS( 30 ) ) )
TTDBLK( ISPACE ) = ( ( PNEXT = *TTDBLK( 30 ), NRTGTYP( 30 ) ),
    ( PTGTPL = *STOBLK( 30 ), NOTGTPL( 30 ) ) )
UMPDAT( ISPACE ) = ( ( MISSILES( 27 ), AMMOTYPE( 3 ), ENGNITTYP( 30 ) ),
    ( PTGTSB = *SB( 30 ), PFU( 30 ) ) )
WAVEBLK( ISPACE ) = ( ( PNEXT = *WAVEBLK( 30 ), NRWAVE( 30 ) ),

```

```

(PGTTYPE=>TARGETBLK(30),NOTGTYP(30)),
STARTTIME(SPACE),
DURATION(SPACE))
WINGMAN(1SPACE)=((PNEXT=>WINGMAN(30),PSB=>SB(30)))
/MACRO
$COMMONAAPK$
COMMON/AAPK/AAPK(10)
$COMMONACFRAG$
COMMON/ACFRAG/NSECTOR,PTGTHEX,PADRCG,PHEXENT,TATREND,PHEXEXT
$COMMONAFM$
COMMON/AFM/JINX
$COMMONAGPD$
COMMON/AGPD/AGPD(30)
$COMMONAGPK$
COMMON/AGPK/AGPK(286)
$COMMONBLKTYP$
COMMON/BLKTYP/KEBASE,NBLKTP,LNGBLK(500)
$COMMONCLOCK$
COMMON/CLOCK/ISTABL,NRSRV,ISTAB(1)
$COMMONCOMOUT$
COMMON/COMOUT/LBL(10,3),LBLCT
$COMMONCOMPTR$
COMMON/COMPTR/
+ PHXTOP, PTRBLUE, PTRRED, PBLTGT, PRDTGT, PTRC2,
+ PBLUPL, PREDPL, PTRPL, PLASTC, PLASTS, LSIDE, LHEX,
+ PTRCDB, PTRWDB, PTRRDB, PTRTTDB, BLFZN,
+ IGMSRT, IDYSRT, PTRDATA, PTUPORD, PREDGEE, ISIZE
$COMMONCOMSCS$
COMMON/COMSCS/LMASK, RMASK, INITCEL, TREETOP, ENDTIME, INCTIME,
+ GTIME, CELMARK, OUTIME, NVFEL, LMAX, CELMINT, LPTR(20)
$COMMONCRCCOM$
COMMON/CRCCOM/IEVENT, PSB, PADR, ITYPE, ISIDE
$COMMONCSTK$
COMMON/CSTK/IPSTK(300), L, LLMAX, NSTK
$COMMONDATA$
COMMON/DATA/PTRDIAT
$COMMONDEBUG$
COMMON/CHECK/ICHPTR(10), LASTVAL(10)
COMMON/DEEUG/DBGA, DBGB, DBS, DEGD
COMMON/DLFLGS/ IDEBUG, IDUMP, IDATFLG, ISTOP, IRECOVR, ITRACE, IRLSE
COMMON/XTRACE/ ITRPTR, TRACIR(50), TRAPTR, TRAPDS(30),
+ TIMEPDS(30), ICOUNT(320), PDSFLAG
COMMON/STEXT/ SEGTXT(320), NOSEG, SEGFLAG(320), SEGTIME(320),
+ DBGFLAG(320)
COMMON/BL/DBG/IBLKHD, IBLKST, IBLKAL, IBLKRL, NELKAL(500),
+ NBLKRL(500)
INTEGER SEGTXT, SEGFLAG, SEGNUM, DBGFLAG
INTEGER TRACIR, TRAPTR, TRAPDS
$COMMONF$
COMMON/FS/PTRFS, PTRRSL, MXSPCE, ID, MXBLKR, LGARB(20)
$COMMONHALT$
COMMON/HALT/IFSTOP, CPULIM, CELTIME, LEVEL(20),
+ GTIMLIM, NOEVNTS, NOEVLIM
$COMMONINITPTR$
COMMON/INITPTR/PTRADR, PTRC2, PTRSDB, PTRFEL, PTRACO, PTRCOM,
..

```

```

+ PTRSTAT, PTRSEE, PTRSUB, PTRORD, MYTYPE, MYSIDE
$COMMONINTMSG$
COMMON/INTMSG/MESSAGE, PTGT$B, ITYPE, PINTLST, PINT$B, PTGTBLK
$COMMONIDDEV$
COMMON/IDDEV/IN, NUOIL, ND, IOUT, I HOLD(2), IFETCH(2), IBREAK, NOHOLDS
$COMMONJJOPT$
COMMON/JJOPT/IOF
$COMMONLIMITS$
COMMON/LIMITS/LOWER, LUPPER
$COMMONMASK$
COMMON/MASK/IL, IR, ILM
$COMMONMODVAR$
COMMON/MODVAR/INCDNT, NEHMEN, LASSEN, TIME, MSG, PTRGOOD, RSEED
$COMMONMXMIS$
COMMON/MXMIS/MX$UP(3, 3)
$COMMONORD$
COMMON/ORD/TBAS, TCOM
$COMMONPATH$
COMMON/PATH/LASTP, LREP, LASTR(15)
$COMMONPERCEEV$
COMMON/PERCEEV/X$EER, Y$EER, X$EEN, Y$EEN, RANGE, BEARING, MASK
$COMMONSAMPK$
COMMON/SAMPK/SAMPKA(3, 3), SAMPKB(3, 3)
$COMMONSAMPTR$
COMMON/SAMPTR$PMYDATA, PDIL, PPPINFO, PDINFO, POINFO, P$AT,
PPAL, P$U
+
$COMMONSECTOR$
COMMON/SECTOR/PDAT, XD, YD, VX, VY, IFLAG, BEARING, VELOCITY, $HEAD,
RELHEAD, CLINE, OFF, X, Y
+
$COMMONSEMINFO$
COMMON/SEMINFO/IVALUE(20), VALUE(20), IFLAG
$COMMONSPACE$
COMMON/SPACE/BLANK, ISPACE(1)
$COMMONSPACE100$
COMMON/SPACE/BLANK, ISPACE(10000)
$COMMONSPACE50$
COMMON/SPACE/BLANK, ISPACE(50000)
$COMMONSPSTAT$
COMMON/SPSTAT/ICTGIM(20), ICTREL(20)
$COMMONSTATBD$
COMMON/STATBD/PFLTYP(1), PTRMUN, PTR$TRT, PTR$END, PTR$XT, PTR$AB,
+ NUMTGT, PTGT$B, PGNDTGT, NUMAC, LEGSTA, INTSTA, IALTCNG, NOX$REF,
+ LNDSTA, IORBSTA, IAIRCOM, IGNDATK, JAMSTA, FUEL, ALTUDE, $PEED, DIRECT
$COMMONTHRPLN$
COMMON/THRPLN/PFOREST, PNOVAL
$COMMONTRACK$
COMMON/TRACK/PFADR, ISIDE
$COMMONTYPES$
COMMON/TYPES/IARRAY(3000)
$DIMEOLEV$
DIMENSION SPACE(1)
EQUIVALENCE( ISPACE(1), SPACE(1))
LEVEL 2, ISPACE, SPACE, BLANK
$IMPLICIT$
IMPLICIT INTEGER(H,P), LOGICAL(Z)
..

```

NOSED
NOSED = 320
/END
..

APPENDIX E
THE DYNAMIC EVENT SCHEDULING ALGORITHM

An event stepped simulation, such as TAC REPELLER, is controlled by one or more time sorted lists of event notices, where each event notice represents the occurrence of some event. At the start of the simulation, all the notices in the event lists correspond to exogenous events, i.e., events which are externally generated and act as a driving function for the simulation. As the simulation progresses, time is incremented as each event notice in the event lists becomes current. An event notice initiates a computational process which may generate additional event notices. In this way the simulation continues. The simulation stops whenever the event lists are exhausted or some other specified termination condition occurs.

The passage of time between events is generally very irregular. In fact, consecutive events may occur at the same instant of time, or there may be a very large time interval between them. Event stepped simulations are computationally efficient because of the ability of the event scheduling mechanism to initiate computation only at those times at which something is going on in the simulation.

The fundamental problem in an event stepped simulation is to devise an efficient way of locating the next event notice in time sequence. This can be done by keeping an event list in time sorted order so that the next event notice is always on the top of the list.

The algorithm for maintaining an event list must be chosen so that it is computationally efficient to 1) remove the event notice with the earliest time from the event list and 2) insert a new event notice into the list.

In TAC REPELLER the event notices for dynamically scheduled events are maintained in a quasi-sorted order using a bifurcated arborescence called a leftist tree (so called because it leans to the left, i.e., there are predominately more links to the left than to the right). The algorithm for manipulating such a tree is such that the top cell of the tree (the root) is always guaranteed to be the cell with the earliest scheduled event time,

even though all other cells are only in quasi-sorted order. Despite some algorithmic complexity, the leftist tree software is very efficient.

Each event notice cell contains the scheduled event time, a path distance value, a pointer to a left subtree, and a pointer to a right subtree. The space requirements are $O(n)$, where n is the number of event notices in the event list.

Since the scheduled event time is the quantity upon which sorting is based, it will be referred to as the KEY in the following discussion. The path distance D is the minimum path length from the node to a leaf of the tree; because of the way in which the tree is constructed, this minimum length path will always be a rightmost path. The pointer to the left subtree will be denoted by LP, and the pointer to the right subtree will be denoted by RP. The pointer to a leaf in the tree will be assigned a value of 0. The leaf does not actually contain any explicit information and therefore leaves are not actually represented in memory. We may refer to the KEY and D quantities for the cells comprising the roots of the left and right subtrees by KEY(LP), $D(LP)$, KEY(RP), and $D(RP)$. We adopt the convention that $KEY(0) = \infty$, and $D(0) = 0$.

The leftist tree may be defined by listing the properties of the KEY and D fields for each cell P :

- (1) $KEY(P) \leq KEY(LP(P))$
- (2) $KEY(P) \leq KEY(RP(P))$
- (3) $D(P) = D(RP(P)) + 1$
- (4) $D(LP(P)) \geq D(RP(P))$

These properties ensure that any path from the root to a leaf traverses the event notices in ascending time order. Thus, the root always contains the next scheduled event notice.

Removal of the root of the leftist tree (i.e., removal of the next scheduled event notice) requires a constant time. However, it must be followed by a merging of the two subtrees below the root before any other operations are executed on the tree. Merging of the subtrees (both of which are themselves leftist trees) is the most expensive operation. In the worst

case this requires m merge steps followed by m interchange steps, where
 m = least integer not less than $\log^2(n)$
 n = number of nodes (event notices) in the tree.

The subtree merging is not required if the right subtree is vacuous, and is trivial if the left subtree is vacuous. Thus, merging of leftist trees requires time of $O(\log n)$ in the worst case. Insertion of a new cell into the tree also requires $O(\log n)$ in the worst case. In the best case, merging and insertion require a constant time. Furthermore, the software for merging and insertion is identical so that only a single routine is required for all tree manipulations. The overhead for small n is reasonable and the method is very efficient for large n .

The algorithm for merging two leftist subtrees P and Q utilizes a stack for saving the nodes which are visited during the tree traversal. The algorithm is:

```

*IF(KEY(SUBTREE P) IS GREATER THAN KEY(SUBTREE Q)) THEN
  *INTERCHANGE SUBTREES P AND Q
*IF (SUBTREE P IS VACUOUS) THEN
  *SUBTREE P = SUBTREE Q
*ELSE
  *MERGE SUBTREE Q INTO SUBTREE P
  *SUBTREE X = P
  *DO WHILE(SUBTREE Q EXISTS)
    *QKEY = KEY(Q)
    *TRAVERSE SUBTREE X ALONG THE RIGHTMOST LINKS, COMPARING KEY
      AT EACH NODE OF X WITH QKEY, SO AS TO LOCATE THE INSERTION
      POINT FOR Q ACCORDING TO ASCENDING KEY VALUE. SAVE THE
      NODES OF X WHICH WERE TRAVERSED ON THE STACK.
    *BREAK SUBTREE X AT THE INSERTION POINT, SAVING THE SUBTREE
      OF X BELOW THE INSERTION POINT AS SUBTREE T.
    *APPEND SUBTREE Q TO SUBTREE X AT THE INSERTION POINT, USING
      THE LEFT LINK IF IT IS NOT ALREADY IN USE: OTHERWISE USE THE
      RIGHT LINK.
    *SUBTREE X = Q
    *SUBTREE Q = T
  *DO(FOR ALL NODES SAVED ON STACK DURING TRAVERSE, STARTING AT LAST
    INSERTION POINT AND ENDING AT THE ROOT)
    *IF(DISTANCE TO LEAF FROM LEFT SUBTREE SHORTER THAN FOR RIGHT
      SUBTREE) THEN
      *INTERCHANGE SUBTREES
    *CALCULATE DISTANCE FOR PARENT NODE OF SUBTREES

```

An example of event sorting using the leftist tree algorithm will now be given. We start with the leftist tree shown in Figure E-1(a). The event notices are numbered with the integers, where the integer value is the scheduled time for the event notice.

The next scheduled event 1 is removed. This splits the original tree into two subtrees as shown in Figure E-1(b). We must merge subtree 3 into subtree 2. This is done by traversing subtree 2 along the rightmost path, until an insertion point is found for subtree 3. This insertion point will be between nodes 2 and 5. A subtree with root 5 is detached and the subtree 3 is appended to subtree 2 at the insertion point. This produces the configuration shown in Figure E-1(c).

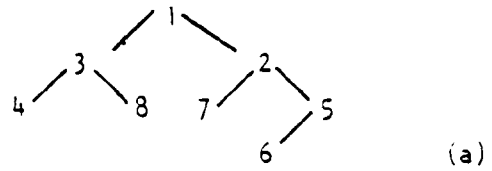
Now subtree 5 must be merged into subtree 2. Subtree 2 is traversed along the rightmost path until an insertion point for subtree 5 is found. This insertion point will be between nodes 3 and 8 in Figure E-1(c). A subtree with root node 8 is detached and subtree 5 is appended to subtree 2 at the insertion point. This produces the configuration shown in Figure E-1(d).

Next subtree 8 must be merged into subtree 2. Subtree 2 is traversed along the rightmost path until an insertion point for subtree 8 is found. This insertion point is after node 5 in Figure E-1(d). Since no right subtree exists below node 5 at this time, no subtree can be detached, and subtree 8 is simply appended as the right subtree for node 5. This produces the configuration shown in Figure E-1(e).

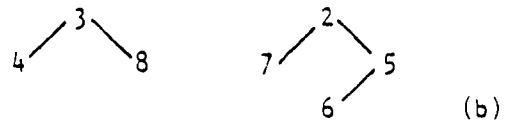
Subtree merging has now been completed. Now the nodes traversed during the merging process must be traversed in reverse order and subtree interchanges made as necessary to produce a leftist tree (note that the tree in Figure E-1(e) is a rightist tree). The subtree with root node 5 is already in leftist form so no action is required.

The subtree with root node 3 is not a leftist tree, therefore interchange the left and right subtrees below node 3. This produces the leftist subtree with root node 3 shown in Figure E-1(f).

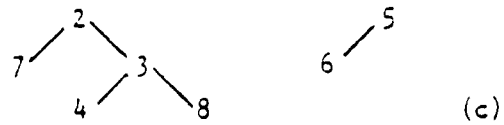
GIVEN



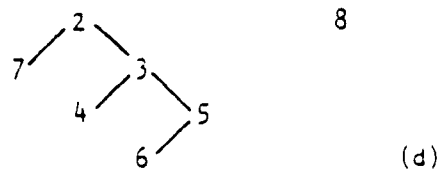
DELETE 1



MERGE 3 INTO 2
REQUIRES DETACH OF 5



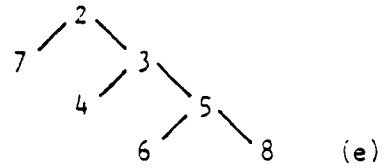
MERGE 5 INTO 2
REQUIRES DETACH OF 8



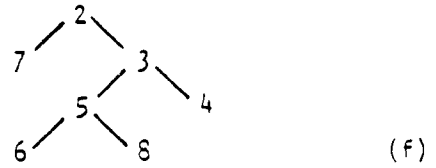
3802/78W

Figure E-1. Example of Merging of Leftist Trees

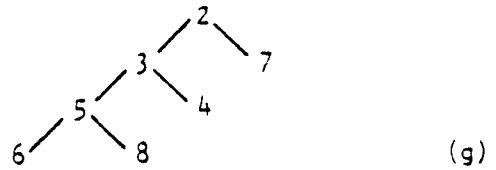
MERGE 8 INTO 2.



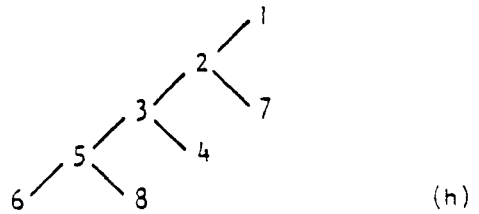
INTERCHANGE 4 AND 5



INTERCHANGE 7 AND 3



ADD 1



3802/75W

Figure E-1. Example of Merging of Leftist Trees (Continued)

The subtree with root node 2 is not a leftist tree, therefore interchange the left and right subtrees below node 2. This produces the leftist subtree with root node 2 shown in Figure E-1(g). Since node 2 is the root of the entire tree, the entire tree is now in leftist form. This completes the merging process.

Now let us add node 1 back into the tree of Figure E-1(g). Since node 1 corresponds to an earlier time than any node of the leftist tree into which it is being merged, we see that node 1 becomes the root node of a new tree. This new leftist tree is shown in Figure E-1(h). Note that its form is radically different from the original tree in Figure E-1(a).

This example represents the worst case situation for merging two subtrees.

APPENDIX F
MADEM EVENT CODE DEFINITIONS

<u>EVENT CODE</u>	<u>MESSAGE CODE</u>	<u>SCHEDULING SUBROUTINE</u>	<u>EVENT</u>
12691111		BADMOVE	Schedule CRC Assignment
12691111		CRCLOSS	Schedule Assignment
12691111		CRCKIL	CRC Assignment
12691111		INT2CRC	Schedule CRC Assignment
12691111		NEWMOVE	Schedule CRC Assignment
14651250		GNDLOOK	Target found, Schedule Attack
25120001	1312	INTASIN	Request interceptor launch
25120001	1510	INTASIN	Message to interceptor of assignment
25120001	2731	BTNASIN	Message BTN assignment
25340001	2751	AMMOCHK	Schedule common BTN (Firing Support)
25390001	1310	INTRFLY	Request orders from CRC
25390001	1340	FUELCHK	Schedule commo CRC, Return to Airbase
25390001	1984	COMMAND	Message to tower to land
25390001	1984	INTRFLY	Request landing from Airbase
25620001	1330	INTFIND	Message to CRC reporting direction
25650001	1300	FLYSEE	Schedule interceptor available, target dead message
2565001	1312	CRCTRAK	Send scramble message
25650001	1320	AIRTHNK	Schedule msg. to live CRC of availability
25650001	1330	CRC2INT	Schedule msg. to CRC can't accept assignment
25690001	2	BNCMDPR	commo CRC not ready
25690001	2	BNNOTRD	Schedule commo CRC not ready

MADDEM EVENT CODE DEFINITIONS

<u>EVENT CODE</u>	<u>MESSAGE CODE</u>	<u>SCHEDULING SUBROUTINE</u>	<u>EVENT</u>
25690001	3	BNPONBB	Schedule commo CRC, ready
25690001	1310	CRC2INT	Schedule interceptor available message to CRC
25690001	1320	AIRTHNK	Schedule msg to CRC of new target
25690001	1340	DOGTHNK	Message to CRC of unavailability
25690001	1500	BADMOVE	Schedule break off message to interceptor
25690001	1520	BADMOVE	Schedule msg to interceptor of new vector
25690001	1520	INT2CRC	Schedule message to interceptor to fly towards CRC (Return to)
25690001	1600	BNCONLS	Schedule commo CRC target lost
25690001	1600	BNCMDPR	Commo CRC loss of sight of target
25690001	1600	BOCTINK	Schedule commo CRC target unseen
25690001	1620	BNCONTC	Schedule commo CRC consider track change
25690001	2752	BYPONRL	Schedule commo BTN reload
25690001	2761	BYHEDUP	Schedule commo BTN no chance
25690001	2761	BNLALLE	Schedule commo superior, no chance
25690001	2762	BYCONLS	Track loss for BTN
25690001	2762	BYNWTRK	Schedule commo BTN track change
25690001	2762	BYCONTC	Schedule commo BTN track change
25690001	2762	BTRYTNK	Schedule commo BTN target not sighted
25690001	2780	BYPONER	Schedule engagement results for BTN
25690001	2790	BYPONFD	Schedule commo BTN flight dead
25690001	2831	ALLOBAT	Btry engage command

MADAM EVENT CODE DEFINITIONS

<u>EVENT CODE</u>	<u>MESSAGE CODE</u>	<u>SCHEDULING SUBROUTINE</u>	<u>EVENT</u>
25690001	2832	BNPONEP	Schedule commo Btry cease fire
25690001	2832	DECRALO	Cease fire msg for battery
25690001	2832	DROPP0S	Schedule commo btry ceasefire
25690001	2832	DROPPS2	Schedule commo btry ceasefire
25690001	2834	ALLOBAT	Btry coverage update
25690001	2834	DECRALO	Coverage update message
25690001	2835	BNCONHD	Schedule commo Btry engagement data update
25930001	1700	GOGETEM	Call CRC to notify of takeoff
31691000		AIRTHNK	Schedule dogfite
34342950		ENGAGE	Schedule engage Btry fire
34342951		ENGAGE	Schedule engage Btry fire
34692900		ALLOPAT	Schedule engage Btry (Lockon)
34692900		ALLOFU	Schedule engage Btry (Lockon)
34692900		BYTKCHK	Schedule engage Btry (Lockon)
34692901		BYPONER	Schedule engage to fire again
34692901		PTPONER	Schedule engage to fire again
34692950		BYPONTM	Schedule engage Btry fire
39391313		FLITE	Schedule next fly
39931313		GOGETEM	Schedule launch of interceptor
39931313		TOWER	Schedule fly flight take off
62141100		ATKASES	Schedule death perception
62311100		DOGFITE	Schedule death perception
62391100		SHRKILL	Schedule ptrgod to perceive death
62391313		FLITE	Schedule Naybor flying plane
62391380		COMMAND	Schedule Naybor ECM on
62391390		COMMAND	Schedule Naybor ECM OFF

MADEM EVENT CODE DEFINITIONS

<u>EVENT CODE</u>	<u>SCHEDULING SUBROUTINE</u>	<u>EVENT</u>
62931100	TOWER	Schedule naybor others notice landing
62951100	UMPIRE	Schedule naybor flight death
65141100	SAMWYPE	Percept superior subordinate dead
65141120	ATTACK	Schedule perception of attack results
65250000	COMMO	Schedule percent reciever (msg)
65311001	DOGFITE	Schedule target to perceive attack
65391350	COMMAND	Schedule rendevous perception of all flights in formation
65391400	COMMAND	Schedule ground target perception
65620000	NAYBOR	Schedule perception of unit.
65621100	DESTROY	Notify CRC of Btn death
65651100	KILFLIT	Notify CRC of subordinate death
65651400	FLYSEE	Resume ground attack
65691400	AIRTHNK	Schedule resume ground attack
67671234	MADEM	Schedule red commander to begin plan of Raid.
67674141	THTRPLN	Plan next raid
69142791	SAMWYPE	Ponder superior subordinate dead
69142793	SAMWYPE	Ponder Btry superior dead
69311905	DOGFITE	Schedule dogfite outcome ponder
69342890	RELOAD	Schedule ponder Btry reload
69391375	FLY	Schedule ponder air combat
6965001	SAMSEE	Schedule ponder message
69651175	FLYSEE	Schedule interceptor message ponder
69651375	FLYSEE	Schedule Air combat ponder

MADEM EVENT CODE DEFINITIONS

<u>EVENT CODE</u>	<u>SCHEDULING SUBROUTINE</u>	<u>EVENT</u>
69651400	CRCEVNT	Schedule CRC ponder
69651410	CRCEVENT	Movement event code
69651420	CRCEVNT	Loss of sight
69651460	CRCSEE	Message from Airbase
69651470	CRCSEE	Message from interceptor
69651480	CRCSEE	Message from BTN
69652790	SAMSEE	Schedule ponder flight death
69652800	BYTKCHK	Ponder track move
69652805	BYTKCHK	Ponder track lost
69692700	ACCEPT	Ponder CE track info digest
69692700	BNLALLE	Schedule digest event
69692700	FILERUP	Schedule ponder digest
69692700	NEWPERC	Schedule ponder CE (track info digest)
69692700	READIL	Schedule ponder Btry digest
69692700	SAMATON	Schedule ponder Sam (Track info digest)
69692700	SDIGEST	Ponder CE Digest track info
69692705	CHKCOV	Ponder BTN delayed action
69692705	SEEKENG	Ponder BTN delayed action
69692705	SEEKTAC	Ponder BTN delayed action
69692710	BYNWTRK	Ponder Btry last chance
69692710	BTHEDUP	Schedule new last chance event
69692710	BYCONHD	Schedule ponder Btry last chance
69692710	BTRYTNK	Last chance event
69692710	BNPONBB	Schedule ponder BTN last chance event
69692710	CHKLAST	Ponder BTN last chance

MADDEM EVENT CODE DEFINITIONS

<u>EVENT CODE</u>	<u>SCHEDULING SUBROUTINE</u>	<u>EVENT</u>
69692710	PREPAFU	Schedule ponder Btry last chance
69692710	SETASSN	Ponder BTN chance gone
69692715	ACCEPT	Schedule expected sighting
69692715	BTRYTNK	Reschedule expected sighting event
69692720	CHKCOV	Schedule opportunity knocks event
69692720	BNPONBB	Schedule opportunity knocks event
69692890	ADASASS	Schedule 1st reload/resupply for each fire unit
69692890	BYPONRL	Schedule next reload
69692895	RESUPLY	Schedule ponder Btry resupply
69952790	UMPIRE	Schedule ponder Btry flight death
69952792	NUKBLND	Schedule ponder BTN (subordinate) cut off)
69952794	NUKBLND	Schedule ponder Btry (superior cut off)
69952880	UMPIRE	Schedule ponder Btry engagement results
69992793	CRCDIES	Schedule superior death event for subordinate
93651312	ABSEE	No launch event scheduled. Schedule launch event
93651984	ABSEE	Schedule tower to land flight
93671313	ACFRAG	Schedule takeoff for flight
93931312	GOGETEM	Schedule another flight for tower to launch
93931312	TOWER	Notify commander of flight landing
95343000	ENGAGE	Schedule UMPIRE

APPENDIX G
RANDOM NUMBER GENERATOR CALLS

MADEM uses the random number generator RANF in all of its' MONTE-CARLO actions. The seed for this uniform random number generator is set using the routine RSEED. Both RANF and RSEED are CDC supplied routines.

ALL USES OF RANF AND RSEED IN MADEM

<u>SUBROUTINE</u>	<u>RANF LINE NUMBERS</u>	<u>RSEED LINE NUMBER</u>	<u>COMMENTS</u>
AIRTANK	70	70	R
ATTACK	81	81	R
CANDTGT	75	75	R
CLIST	--	43	R
CRCEVENT	98	98	R
CRCTRAK	83	83	R
DETECT	89	89	R
DOGFITE	72	72	R
FILERUP	28	28	R
GNDLOOK	35	35	R
NEWMOVE	29	29	R Problem of Error Increases Over Time
OUTPTRS	--	27	R
PTRAND	16, 33, 38	16, 33, 38	R
RELEASE	--	22, 34, 45	(Writes) R
SCHEDUL	38, 50	38, 50	R
SDIGEST	58	58	R
SHRKILL	30	30	R
VMPIRE	37	37, 37, 39	RSEED=RANCRSEED (SET)

R = reference only
SET = RSEED set in the subroutine

APPENDIX H
MADEM SUBROUTINE REFERENCE LISTS
AND CALLING HIERARCHIES

1. Preprocessor

PAGE 1

LIST OF SUBROUTINES - MADEM

1. ABQUEUE
2. ABVSCOR
3. ACFRAG
4. ADASASS
5. AOOBLOK
6. ADDCHR
7. ADUMP
8. APCELI
9. APCEL2
10. ASSIGN
11. ATTACK
12. AVAILBL
13. BDALT
14. BOLEX
15. BDLRK
16. BOPARS
17. BLKDAT
18. CANTGT
19. CARD
20. CENTER
21. CHGEN
22. CLIST
23. CLIST2
24. CLOSCOR
25. CODE01
26. CODE03
27. CODE05
28. CODE18
29. COMMO
30. CONTROL
31. CORBQUN
32. CREATE
33. CRFLTML
34. DBGREAD
35. DECDMS
36. DELADD
37. DGTSHX
38. DISPA8Q
39. DISPACD
40. DISPACL
41. DISPACR
42. DISPADS
43. DISPAQD
44. DISPDAT
45. DISPFDB
46. DISPFLT
47. DISPFMF
48. DISPPAF
49. DISPPAY
50. DISPPRO
51. DISPPYB
52. DMSDEC

53. DOGFITE
54. DROPBLK
55. ENGAGE
56. ENTRYP
57. ENTSTAT
58. EOF
59. ERROR
60. EXITP
61. EXTSCN
62. FELDEL
63. FETCH
64. FINDBLK
65. FINDFLT
66. FLTGEOM
67. FLY
68. FORMTGT
69. FSDUMP
70. FSINIT
71. GETHEX
72. GETPTRS
73. GIMME
74. HALT
75. HEXAOD
76. HEXCHZ
77. HEXDIST
78. HEXINV
79. HEXMLT
80. HEXMULT
81. HISTORY
82. HLTPNT
83. HOLD
84. HXDGTS
85. HXMLT2
86. ICHECK
87. IJ2HX
88. INIT
89. INITACQ
90. IPJL
91. ISDUMP
92. ISHIFT
93. ITRAP
94. JGESUIT
95. JTJ
96. JUGGLE
97. KOMPARE
98. LACELL
99. LCHLOC
100. LEXAN
101. LINEX
102. LNPLDT
103. LOADPL
104. LOOKUP
105. LRPRES
106. LTREF
107. LTRMRG

108. MADEM
109. MASKER
110. MESSAGE
111. NAYBOR
112. NOWUCIT
113. NXTSYM
114. OPTPTH
115. OTHROAT
116. OUTA
117. OUTPTRS
118. PACK
119. PAGE
120. PELADD
121. PERCEPT
122. PLAN
123. PLANOUT
124. PONDER
125. PTFEF
126. PTRMRG
127. RDCCELL
128. RECCON
129. RECER
130. RECOVR
131. RELEASE
132. RELIST
133. RENDEVU
134. REVISE
135. RTEY
136. RTEP
137. RTER
138. RLABDB
139. RLCORD
140. RLFMAKT
141. RLRAID
142. RLTGTAK
143. RLTGTYP
144. RLWAVE
145. ROUTER
146. SCHEDUL
147. SHTAB
148. SECOND
149. SELECT
150. SEMANT
151. SHUFFLE
152. SNAP
153. SRCHPL
154. TGTGONE
155. TGTLIST
156. THM2PS
157. THTRPLN
158. THX2XY
159. TH2HX
160. TLL2HX
161. TLL2XY
162. TOWER

163. TRACE
164. TRPRNT
165. TRPRNT
166. TRPRRT
167. TTIME
168. TXY2HX
169. TXY24XL
170. TXY2LL
171. UMPIRE
172. UNPACK
173. UNSNAP
174. UOLLOAD
175. WTIPEOUT
176. XSHIFT

PAGE 4

LIST OF FORTRAN LIBRARY ROUTINES - MADEM

1. ALOG.
2. ASIN.
3. ATAN2.
4. ATAN.
5. COS.
6. DECODI.
7. ENDFIL.
8. END.
9. GOTOER.
10. INPBI.
11. INPCI.
12. INPCR.
13. INPFI.
14. ITOJ.
15. OUTBI.
16. OUTCI.
17. OUTCR.
18. QINTRY.
19. RANDOM.
20. REWIND.
21. SIN.
22. STOP.
23. TAN.
24. TAPE6#
25. XTOI.

SUBROUTINE REFERENCE LIST - MADEM

PAGE 6

1. ABQUEUE	CALLS: ENTRYP GIMME ADDBLOK FINDBLK EXITP	CALLED BY: CODE05 CODE03
2. ABVSCOR	CALLS: ENTRYP CLOSCOR GIMME ADDBLOK EXITP	CALLED BY: REVISE
3. ACFRAG	CALLS: ENTRYP GIMME CRFLTML ADDBLOK OPTPTH RELEASE FLTGEOM HEXDIST DELAOD EXITP	CALLED BY: SCHEDUL
4. ADASASS	CALLS: ENTRYP GIMME FINDBLK GOTOER. DELAOD ADDBLOK EXITP	CALLED BY: CODE03
5. ADDBLOK	CALLS: ENTRYP EXITP	CALLED BY: UOLLOAD T3TLIST SEMANT OTHRDAT NOWUCIT INITACQ FINDFLT CRFLTML CANOTGT ADASASS ACFRAG ABVSCOR ABQUEUE
6. ADDCHR	CALLS:	CALLED BY:

	ISHIFT TAPE6# OUTCI.	LEXAN
7. ADUMP	CALLS: OUTCI. OUTCR.	CALLED BY: HALT RECCON
8. APCEL1	CALLS: ISHIFT TAPE6# OUTCI.	CALLED BY: LRKPRS
9. APCEL2	CALLS: ISHIFT TAPE6# OUTCI.	CALLED BY: LRKPRS
10. ASSIGN		CALLED BY: SELECT
11. ATTACK		CALLED BY: SELECT
12. AVAILBL	CALLS: ENTRYP PELADD RELEASE EXITP	CALLED BY: THTRPLN
13. BDALT		
14. BOLEX		
15. BDLRK		
16. BDPARS		
17. BLKDAT		
18. CANDTGT	CALLS: ENTRYP FINDBLK CLOSCOR JGESJIT FORMTGT GIMME ADDBLOK TGTGONE PTREE RANDOM. PELADD EXITP	CALLED BY: THTRPLN
19. CARD	CALLS:	CALLED BY:

	INPCI. EOF TAPE6# OUTCI.	CHRGEN
20. CENTER	CALLS: ENTRYP EXITP	CALLED BY: TXYZHXL TLLZHX
21. CHRGEN	CALLS: CARD TAPE6# OUTCI.	CALLED BY: LEXAN EXTSCN
22. CLIST	CALLS: PAGE LNPLT MESSAGE RITEP RITER RITEI TAPE6# OUTCI. OUTCR. CLIST2	CALLED BY: HALT HEXCHZ FSOUMP FINDBLK
23. CLIST2	CALLS: PAGE LNPLT MESSAGE RITEI RITER TAPE6# OUTCI. OUTCR. RITEP	CALLED BY: CLIST
24. CLOSCOR	CALLS: ENTRYP HEXDIST EXITP	CALLED BY: CANDTGT A9VSCOR
25. CODE01	CALLS: ENTRYP GIMME RITEP EXITP	CALLED BY: SEMANT
26. CODE03	CALLS: ENTRYP SRCHPL CREATE LOADPL RITEP LNPLT	CALLED BY: SEMANT

	ABQUEUE ADASASS EXITP	
27. CODE05	CALLS: ENTRYP SRCHPL FINDBLK GETHEX RITEI RITEP LNPLDT UOLLOAD ABQUEUE TGTLIST INITACO HISTORY EXITP	CALLED BY: SEMANT
28. CODE18	CALLS: GETHEX GIMME PACK UNPACK UOLLOAD RITEP RITEI LNPLDT TGTLIST HISTORY	CALLED BY: SEMANT
29. COMMO		CALLED BY: SELECT
30. CONTROL	CALLS: ENTRYP LTREE SELECT SNAP UNSNAP RELEASE SECOND HLTPNT EXITP	CALLED BY: MADEM
31. CORBOUN	CALLS: ENTRYP GIMME THH2PS TXY2HXL HEXDIST SIN. COS. LINEX OPTPTH	CALLED BY: THTRPLV

	EXITP	
32. CREATE	CALLS: ENTRYP GIMME EXITP	CALLED BY: SRCHPL CRFLTML CODE03
33. CRFLTML	CALLS: ENTRYP CREATE GIMME ADDBLOK HISTORY EXITP	CALLED BY: ACFRAG
34. OBGREAD	CALLS: OUTCI. INPCI. EOF DECOOI.	CALLED BY: MADEM
35. DECDMS		
36. JELADD	CALLS: ENTRYP MESSAGE TRACE RECER GIMME SNAP LTRMRG EXITP	CALLED BY: THTRPLN ADASASS ACFRAG MADEM
37. DGTSHX	CALLS: ITOJ.	
38. DISPA8Q	CALLS: OUTCI.	CALLED BY: DISPDAT
39. DISPACD	CALLS: OUTCI.	CALLED BY: DISPFLT DISPDAT
40. DISPACL	CALLS: OUTCI.	CALLED BY: DISPACR
41. DISPACR	CALLS: OUTCI. DISPACL	CALLED BY: DISPDAT
42. DISPADS	CALLS: OUTCI.	CALLED BY: DISPDAT
43. DISPAQD	CALLS: OUTCI.	CALLED BY: DISPFLT

		DISPDAT
44. DISPOAT	CALLS:	CALLED BY:
	OUTCI.	MADEN
	DISPAOS	
	DISPFDB	
	DISPFLT	
	DISPACD	
	DISPPAF	
	DISPPRO	
	DISPAQD	
	DISPABQ	
	DISPACH	
45. DISPFDB	CALLS:	CALLED BY:
	OUTCI.	DISPOAT
	DISPFMF	
46. DISPFLT	CALLS:	CALLED BY:
	OUTCI.	DISPFMF
	DISPPAY	DISPOAT
	DISPAQD	
	DISPACD	
	DISPPRO	
47. DISPFMF	CALLS:	CALLED BY:
	OUTCI.	DISPFDB
	DISPFLT	
48. DISPPAF	CALLS:	CALLED BY:
	OUTCI.	DISPOAT
	DISPPYB	
49. DISPPAY	CALLS:	CALLED BY:
	OUTCI.	DISPFLT
	DISPPYB	
50. DISPPRO	CALLS:	CALLED BY:
	OUTCI.	DISPFLT
		DISPOAT
51. DISPPYB	CALLS:	CALLED BY:
	OUTCI.	DISPPAY
		DISPPAF
52. DMSDEC	CALLS:	CALLED BY:
	ENTRYP	SEMANT
	EXITP	
53. DOGFITE		CALLED BY:
		SELECT
54. DROPBLK	CALLS:	CALLED BY:
	ENTRYP	NOWUCIT
	RELEASE	

EXITP

55. ENGAGE

CALLED BY:
SELECT

56. ENTRYP

CALLS:

MESSAGE
 RITEI
 RECER
 TAPE6#
 OUTCI.
 ROUTER
 ITRAP
 SECOND

CALLED BY:
 LTRMRG
 UNPACK
 UOLLOAD
 UNSNAP
 TXY2HXL
 TXY2HX
 TTIME
 TLL2HX
 THX2XY
 THTRPLY
 THH2PS
 TGTLIST
 TGTGONE
 SRCHPL
 SNAP
 SEMANT
 SELECT
 SCHTAB
 SCHEDUL
 RLWAVE
 RLGTYP
 RLGTAK
 RLRAT
 RLFMA.T
 RLCORD
 RLABOB
 REVISE
 RENDEVU
 RELIST
 RELEASE
 PTREE
 PLANOUT
 PLAN
 PELADD
 PACK
 OUTA
 OTHROAT
 OPTPTH
 NOWUCIT
 LTREE
 LOADPL
 LINEX
 KOMPARE
 JUGGLE
 JTJ
 JGESUIT
 INITACQ
 INIT
 IJ2HX

HKOGTS
 HOLD
 HLPNT
 HISTORY
 HEXMULT
 HEXMLT
 HEXINV
 HEXCHZ
 HEXADD
 GIMME
 GETPTRS
 GETHEX
 FSINIT
 FSDUMP
 FORMTGT
 FLTGEOM
 FINDFLT
 FINDBLK
 FETCH
 DROPBLK
 OMSDEC
 DELADD
 CRFLTML
 CREATE
 CORBOUN
 CONTROL
 CODE05
 CODE03
 CODE01
 CLOSCOR
 CENTER
 CANOTGT
 AVAILBL
 ADDBLK
 ADASASS
 ACFRAG
 ABVSCOR
 ABQUEUE
 MADEM

57. ENTSTAT

CALLS:
 TAPE6#
 OUTCI.

CALLED BY:
 HALT
 FSDUMP

58. EOF

CALLED BY:
 JBGREAD
 CARD

59. ERROR

CALLS:
 TAPE6#
 OUTCI.

CALLED BY:
 LRKPRS

60. EXITP

CALLS:
 SECOND
 MESSAGE

CALLED BY:
 LTRMRG
 UNPACK

RITEI
RECER
TAPE6#
OUTCI.
ITRAP
ICHECK

PAGE 14

UOLLOAD
UNSNAP
TXY2HXL
TXY2HX
TTIME
TLL2HX
THX2XY
THTRPLN
THM2PS
TGTLIST
TGTGONE
SRCHPL
SNAP
SEMANT
SELECT
SCHTAB
SCHEDUL
RLWAVE
RLTGTYP
RLTGAK
RLRAID
RLFMAKT
RLCORD
RLABDB
REVISE
RENOEUV
RELIST
RELEASE
PTREE
PLANOUT
PLAN
PELAOD
PACK
OUTA
OTHRDAT
OPTPTH
NOWUCIT
LFREE
LOADPL
LINEX
KOMPARE
JUGGLE
JTJ
JGESUIT
INITACQ
INIT
IJ2HX
HXDGTS
HOLD
HLTPNT
HISTORY
HEXMULT
HEXMLT
HEXINV
HEXCHZ

HEXADD
 GIMME
 GETPTRS
 GETHEX
 FSINIT
 FSDUMP
 FORMTGT
 FLTGEOM
 FINDFLT
 FINDBLK
 FETCH
 DROPBLK
 DMSDEC
 DELADD
 CRFLTML
 CREATE
 CORRBOUN
 CONTROL
 CODE05
 CODE03
 CODE01
 CLOSCOR
 CENTER
 CANDTGT
 AVAILBL
 ADDBLK
 ADASASS
 ACFRAG
 ABVSCOR
 ABQUEUE
 MADEM

61. EXTSCN	CALLS: CHRGEN IT0J.	CALLED BY: LEXAN
62. FELDEL		CALLED BY: SELECT
63. FETCH	CALLS: ENTRYP INPBI. EXITP	CALLED BY: MADEM
64. FINDBLK	CALLS: ENTRYP ROUTER MESSAGE RITEI CLIST EXITP	CALLED BY: THTRPLN T3TLIST SEMANT OTHRDAT NOWUCIT FORMTGT FINDFLT CODE05 CANDTGT ADASASS

65. FINDFLT	CALLS: ENTRYP HEXDIST FINOBLK KOMPARE GIMME ADDBLOK EXITP	ABQUEUE CALLED BY: FORMTGT
66. FLTGEOM	CALLS: ENTRYP HEXCHZ THM2PS ATAN2. EXITP	CALLED BY: ACFRAG
67. FLY		CALLED BY: SELECT
68. FORMTGT	CALLS: ENTRYP GIMME FINDBLK FINDFLT RELEASE EXITP	CALLED BY: CANOTGT
69. FSDUMP	CALLS: ENTRYP TRACE ENTSTAT CLIST RITER LNPL0T LCMLOC TAPE6# OUTCI. EXITP	CALLED BY: JTJ
70. FSINIT	CALLS: ENTRYP EXITP	CALLED BY: MADEM
71. GETHEX	CALLS: ENTRYP HXDGTS MESSAGE RITEI GIMME SCHTAB EXITP	CALLED BY: TXYZHL SEMANT SCHEDUL NOWUCIT HEXCHZ COOE18 COOE05

72. GETPTRS	CALLS: ENTRYP EXITP	CALLED BY: PLAN
73. GIMME	CALLS: ENTRYP HALT MESSAGE RITEI RITEP EXITP	CALLED BY: UOLLOAD TGTLIST TGTGONE SEMANT REVISE RELEASE PELADD OTHRDAT OPTPTH NOWUCIT INITACO INIT GETHEX FORMTGT FINDFLT DELADD CRFLTML CREATE CORBOUN CODE18 CODE01 CANOTGT ADASASS ACFRAG ABVSCOR ABQUEUE
74. HALT	CALLS: HOLD OUTCI. RITER RITEI TRACE RECER ENTSTAT CLIST PAGE ADUMP ISDUMP ENDFIL. STOP.	CALLED BY: UNPACK ITRAP RECCON SELECT RELEASE PACK OTHRDAT HLTPNT GIMME MADEM
75. HEXADD	CALLS: ENTRYP IT0J. EXITP	CALLED BY: SCHEDUL NOWUCIT HEXMLT HEXCHZ
76. HEXCHZ	CALLS:	CALLED BY:

	ENTRYP TRACE MESSAGE RITEP CLIST HEXADD HEXINV GETHEX EXITP	OPTPTH FLTGEOM
77. HEXDIST		CALLED BY: SCHEDUL FINOFLT CORBOUN CLOSCOR ACFRAG
78. HEXINV	CALLS: ENTRYP ITOJ. EXITP	CALLED BY: HEXCHZ
79. HEXMLT	CALLS: HEXADD ENTRYP HXDGTS MESSAGE RITEI EXITP	CALLED BY: IJ2MX
80. HEXMUL	CALLS: ENTRYP ITOJ. EXITP	CALLED BY: NOWUCIT
81. HISTORY	CALLS: ENTRYP MESSAGE MASKER TAPE6# OUTCI. EXITP	CALLED BY: CRFLTML CODE18 CODE05
82. HLTPT	CALLS: ENTRYP SECOND HALT EXITP	CALLED BY: CONTROL
83. HOLD	CALLS: ENTRYP OUTBI. REWIND. EXITP	CALLED BY: HALT

84. HXDGTS	CALLS: ENTRYP EXITP	CALLED BY: TH2HX THX2XY HEXMLT GETHEX
85. HXMLT2	CALLS: TH2HX IJ2HX	
86. ICHECK	CALLS: OUTCI.	CALLED BY: EXITP
87. IJ2HX	CALLS: HEXMLT ENTRYP ITOJ. EXITP	CALLED BY: TXY2HXL TLL2HX HXMLT2
88. INIT	CALLS: ENTRYP XTOI. GIMME RITEP SNAP EXITP	CALLED BY: MADEM
89. INITACQ	CALLS: ENTRYP GIMME ADDBLOK ALOG. MESSAGE RITEI XTOI. NOWUCIT EXITP	CALLED BY: CODE05
90. IPJL	CALLS: LCMLOC ISHIFT TAPE6# OUTCI.	
91. ISDUMP	CALLS: OUTCI. OUTCR.	CALLED BY: HALT UNPACK PACK
92. ISHIFT		CALLED BY: APCEL2 LACELL APCEL1 ROCELL

ADDCR
 LOOKUP
 LRKPRS
 LEXAN
 PTREE
 JTJ
 IPJL

93. ITRAP	CALLS: HALT	CALLED BY: EXITP ENTRYP THTRPLN
94. JGESUIT	CALLS: ENTRYP THX2XY ATAN2. EXITP	CALLED BY: CANDTGT
95. JTJ	CALLS: ENTRYP MESSAGE RITEI ISHIFT TAPE6# OUTCI. FSDUMP STOP. EXITP	
96. JUGGLE	CALLS: ENTRYP EXITP	CALLED BY: THX2XY
97. KOMPARE	CALLS: ENTRYP UNPACK PACK EXITP	CALLED BY: THTRPLN FINDFLT
98. LACELI	CALLS: ISHIFT TAPE6# OUTCI.	CALLED BY: LRKPRS
99. LCML0C		CALLLED BY: UNPACK RITEP PACK IPJL FSDUMP MADEM
100. LFXAN	CALLS: CHRGEN	CALLLED BY: NXTSYM

	ISHIFT GOTOER. ADDCHR LOOKUP EXTSCN TAPE6# OUTCI.	
101. LINEX	CALLS: ENTRYP EXITP	CALLED BY: OPTPTH CORBOUN
102. LNPL0T	CALLS: TAPE6# OUTCI.	CALLED BY: CLIST2 SEMANT OUTPTRS FSDUMP CODE18 CODE05 CODE03 CLIST
103. LOADPL	CALLS: ENTRYP EXITP	CALLED BY: CODE03
104. LOOKUP	CALLS: ISHIFT TAPE6# OUTCI.	CALLED BY: LEXAN
105. LRXPRL	CALLS: SEMANT TAPE6# OUTCI. R0CELL NXTSYM ERROR ISHIFT APCEL1 APCEL2 LACELL	CALLED BY: MADEM
106. LTR00	CALLS: ENTRYP RELEASE LTRMRG EXITP	CALLED BY: CONTROL
107. LTRMRG	CALLS: ENTRYP EXITP	CALLED BY: LTR00 DELADD
108. MADEM	CALLS: GINTRY.	

RECCON
ENTRYP
INPFI.
DBGREAD
RECOVR
GOTOER.
FETCH
PAGE
LCMLOC
FSINIT
OTHRDAT
DISPOAT
HALT
INIT
LRKPRS
RELIST
DELADD
CONTROL
EXITP
END.

109. MASKER

CALLLED BY:
HISTORY

110. MESSAGE

CALLS:
TAPE6#
OUTCI.

CALLLED BY:
CLIST2
EXITP
ENTRYP
SELECT
RELEASE
PACK
OUTPTRS
JTJ
INITACQ
HISTORY
HEXMLT
HEXCHZ
GIMME
GETHEX
FINDBLK
DELADD
CLIST

111. MAYBOR

CALLLED BY:
SELECT

112. NOWUCIT

CALLS:
ENTRYP
HEXADD
HEXMULT
GETHEX
GIMME
ADDBLOK
FINDBLK
DROPBK

CALLLED BY:
INITACQ

RELEASE
EXITP

113. NXSYPH	CALLS: LEXAN TAPE6# OUTCI.	CALLED BY: LTKPRS
114. ORTPTH	CALLS: ENTRYP GIMME TMM2PS HEXCHZ LINEX EXITP	CALLED BY: CORBOUN ACFRAG
115. OTHROAT	CALLS: ENTRYP GIMME INPCI. TAPE6# OUTCI. RITEP ADDBLOK INPFI. FINOBLK HALT EXITP	CALLED BY: MAOEM
116. OUTA	CALLS: ENTRYP TAPE6# OUTCI. EXITP	
117. OUTPTRS	CALLS: LNPLOT MESSAGE RITEI RITER RITEP UNPACK	
118. PACK	CALLS: ENTRYP LCMLC PAGE MESSAGE RITEI TRACE ROUTER RITEP ISDUMP HALT EXITP	CALLED BY: SEMANT KOMPARE CODE18

119. PAGE	CALLS: TAPE6# OUTCI.	CALLED BY: CLIST2 HALT PACK CLIST MADEM
120. PELADD	CALLS: ENTRYP GIMME PTRMRG EXITP	CALLED BY: TGTLIST CANDTGT AVAILBL
121. PERCEPT		CALLED BY: SELECT
122. PLAN	CALLS: ENTRYP GETPTRS THTRPLN EXITP	CALLED BY: SELECT
123. PLANOUT	CALLS: ENTRYP OUTCI. EXITP	CALLED BY: THTRPLN
124. PONDER		CALLED BY: SELECT
125. PTREE	CALLS: ENTRYP RELEASE ISHIFT EXITP	CALLED BY: TGTGONE CANDTGT
126. PTRMRG		CALLED BY: PELADD
127. ROCELL	CALLS: ISHIFT TAPE6# OUTCI.	CALLED BY: LRKPRS
128. RECCON	CALLS: TAPE6# OUTCI. ADUMP HALT	CALLED BY: MADEM
129. RECER	CALLS: TAPE6# OUTCI. OUTCR.	CALLED BY: HALT EXITP ENTRYP

		DELADD
130. RECOVR		CALLER BY: MADEM
131. RELEASE	CALLS:	CALLER BY:
	ENTRYP	RLWAVE
	MESSAGE	RLTGTYP
	RITEI	RLTGTA
	RITEP	RLRAID
	HALT	RLMAKT
	GIMME	RLCORD
	EXITP	RLABOB
		REVISE
		RELIST
		PTR
		NOWUCIT
		LTR
		FORMTGT
		DROPBLK
		CONTROL
		AVAILBL
		ACFRAG
132. RELIST	CALLS:	CALLER BY: MADEM
	ENTRYP	
	RELEASE	
	EXITP	
133. RENDEVU	CALLS:	CALLER BY: SCHEDUL
	ENTRYP	
	THX2XY	
	TXYZXL	
	EXITP	
134. REVISE	CALLS:	CALLER BY: THTRPLN
	ENTRYP	
	GIMME	
	ABVSCOR	
	RELEASE	
	EXITP	
135. RITEI	CALLS:	CALLER BY:
	TAPE6*	CLIST2
	OUTCI.	HALT
		EXITP
		ENTRYP
		SEMANT
		SELECT
		RELEASE
		PACK
		OUTPTRS
		JTJ
		INITACQ
		HEXMLT

			GIMME GETHEX FINDBLK CODE18 CODE05 CLIST
136. RITEP	CALLS: LCML0C TAPE6# OUTCI.	CALLED BY:	CLIST2 UNPACK SEMANT RELEASE PACK OUTPTRS OTHRDAT INIT HEXCHZ GIMME CODE18 CODE05 CODE03 CODE01 CLIST
137. RITER	CALLS: TAPE6# OUTCI.	CALLED BY:	CLIST2 HALT SEMANT OUTPTRS FSDUMP CLIST
138. RLAB08	CALLS: ENTRYP RELEASE EXITP	CALLED BY:	RLCORD
139. RLCORD	CALLS: ENTRYP RLAB08 RELEASE EXITP	CALLED BY:	RLRAID
140. RLFMAKT	CALLS: ENTRYP RELEASE EXITP	CALLED BY:	RLTGTA
141. RLRAID	CALLS: ENTRYP RLWAVE RLCORD RELEASE EXITP	CALLED BY:	THTRPLN

142. RLGTAK	CALLS: ENTRYP RLFMKT RELEASE EXITP	CALLED BY: RLGTYP
143. RLGTYP	CALLS: ENTRYP RELEASE RLGTAK EXITP	CALLED BY: RLWAVE
144. RLWAVE	CALLS: ENTRYP RLGTYP RELEASE EXITP	CALLED BY: RLRAID
145. ROUTER	CALLS: TAPE6# OUTCI. OUTCR.	CALLED BY: ENTRYP PACK FINOBLK
146. SCHEDUL	CALLS: ENTRYP UNPACK RANDOM. RENDEVU HEXADD GETHEX HEXDIST ACFRAG EXITP	CALLED BY: THTRPLN
147. SHTAB	CALLS: ENTRYP EXITP	CALLED BY: GETHEX
148. SECGND		CALLED BY: EXITP ENTRYP MLTPNT CONTROL
149. SELECT	CALLS: ENTRYP GCTOER. MESSAGE RITEI HALT FELDEL ASSIGN ATTACK COMMO	CALLED BY: CONTROL

DOGFITE
 ENGAGE
 FLY
 NAYBOR
 PERCEPT
 PLAN
 PONDER
 TOWER
 UMPIRE
 TAPE6#
 OUTCI.
 OUTCR.
 EXITP

150. SEMANT

CALLS:

ENTRYP
 GOTOER.
 CODE01
 GIMME
 PACK
 RITEI
 RITEP
 LNPL0T
 CODE03
 GETHEX
 CODE05
 DMSDEC
 TLL2HX
 INPCI.
 INPCR.
 TAPE6#
 OUTCI.
 OUTCR.
 ADDBLOK
 TTIME
 RITER
 FINDBLK
 CODE18
 XTOI.
 SRCHPL
 EXITP

CALLED BY:

LKXPRS

151. SHUFFLE

CALLED BY:

T42HX

152. SNAP

CALLS:

ENTRYP
 EXITP

CALLED BY:

INIT
 DELADD
 CONTROL

153. SRCHPL

CALLS:

ENTRYP
 CREATE
 EXITP

CALLED BY:

SEMANT
 C00E05
 C00E03

154. TGTGONE	CALLS: ENTRYP PTREE GIMME EXITP	CALLED BY: CANOTGT
155. TGTLIST	CALLS: ENTRYP FINOBLK GIMME ADDBLOK PELADD EXITP	CALLED BY: CJ0E18 CJ0E05
156. THH2PS	CALLS: ENTRYP THX2XY EXITP	CALLED BY: OPTPTH FLTGEOM CORBOUN
157. THTRPIN	CALLS: ENTRYP CORBOUN REVISE FINOBLK KOMPARE CANOTGT ITRAP AVAILBL SCHEDUL DELADD PLANOUT RLRAIO EXITP	CALLED BY: PLAN
158. THX2XY	CALLS: ENTRYP HXDGTS JUGGLE EXITP	CALLED BY: THH2PS RENDEVU JGESUIT
159. TH2HX	CALLS: HXDGTS SHUFFLE	CALLED BY: HXMLT2
160. TLL2HX	CALLS: ENTRYP SIN. TAN. TXY2HX CENTER IJ2HX EXITP	CALLED BY: SEMANT
161. TLL2XY	CALLS:	

	SIN. TAN.	
162. TOWER		CALLED BY: SELECT
163. TRACE	CALLS: TRPRRT TRPRMT TRPRNT	CALLLED BY: HALT PACK HEXCHZ FSDUMP DELOAD
164. TRPRMT		CALLLED BY: TRACE
165. TRPRNT	CALLS: TAPE6# OUTCI.	CALLLED BY: TRACE
166. TRPRRT		CALLLED BY: TRACE
167. TTIME	CALLS: ENTRYP EXITP	CALLLED BY: SEMANT
168. TXY2HX	CALLS: ENTRYP EXITP	CALLLED BY: TLL2HX
169. TXY2HXL	CALLS: ENTRYP COS. SIN. XTOI. CENTER IJ2HX GETHEX EXITP	CALLLED BY: RENDEUVU CORBOUN
170. TXY2LH	CALLS: ASIN. ATAN.	
171. UMPIRE		CALLLED BY: SELECT
172. UNPACK	CALLS: ENTRYP LCMLOC RITP ISDUMP HALT EXITP	CALLLED BY: SCHEDUL OUTPTRS KOMPARE CODE18

173. UNSNAP

CALLS:
ENTRYP
EXITP

CALLED BY:
CONTROL

174. UOLLOAD

CALLS:
ENTRYP
GIMME
ADOBLOK
EXITP

CALLED BY:
CODE18
CODE05

175. WIPEDOUT

176. XSHIFT

1. ALOG.	CALLED BY: INITACQ
2. ASIN.	CALLED BY: TXY2LL
3. ATANZ.	CALLED BY: JGESUIT FLTGEOM
4. ATAN.	CALLED BY: TXY2LL
5. COS.	CALLED BY: TXY2HXL CORBOUN
6. DECODI.	CALLED BY: DBGREAD
7. ENDFIL.	CALLED BY: HALT
8. END.	CALLED BY: MADEM
9. GOTOER.	CALLED BY: LEXAN SEMANT SELECT ADASASS MADEM
10. INPBI.	CALLED BY: FETCH
11. INPCI.	CALLED BY: DBGREAD CARD SEMANT OTHRDAT
12. INPCR.	CALLED BY: SEMANT
13. INPFI.	CALLED BY: OTHRDAT MADEM
14. ITOJ.	CALLED BY: EXTSCN

IJ2HX
HEXMULT
HEXINV
HEXA00
DGTSHX

15. OUTBI.

CALLED BY:

HOLD

16. OUTCI.

CALLED BY:

CLIST2
ISDUMP
HALT
ADUMP
DBGREAD
ICHECK
DISPPY8
DISPPRO
DISPPAY
DISPPAF
DISPPMF
DISPFLT
DISPFD9
DISPA00
DISPADS
DISPACR
DISPACL
DISPACO
DISPABQ
DISPDAT
EXITP
ENTRYP
ENTSTAT
ROUTER
RECER
RECCON
NXTSYM
APCEL2
LACELL
APCEL1
ROCELL
CARD
CHRGEN
ERROR
ADDCHR
LOOKUP
LRKPRS
LEXAN
TRPRNT
SEMANT
SELECT
RITER
RITEP
RITEI
PLANOUT

	PAGE
	OUTA
	OTHRDAT
	MESSAGE
	LVPLDT
	JTJ
	IPJL
	HISTORY
	FSDUMP
	CLIST
17. OUTCR.	CALLLED BY:
	CLIST2
	ISDUMP
	ADUMP
	ROUTER
	RECER
	SEMANT
	SELECT
	FSDUMP
	CLIST
18. 21NTRY.	CALLLED BY:
	MADEM
19. RANDOM.	CALLLED BY:
	SCHEDUL
	CANDTGT
20. REWIND.	CALLLED BY:
	HOLD
21. SIN.	CALLLED BY:
	TXY2HXL
	TLL2XY
	TLL2HX
	CORBOUN
22. STOP.	CALLLED BY:
	HALT
	JTJ
23. TAN.	CALLLED BY:
	TLL2XY
	TLL2HX
24. TAPE6#	CALLLED BY:
	CLIST2
	EXITP
	ENTRYP
	ENTSTAT
	ROUTER
	RECER
	RECCON
	NXTSYM

PAGE 35

APCEL2
LACELL
APCEL1
ROCELL
CARD
CHRGEN
ERROR
ADDCHR
LOOKUP
LRKPRS
LEXAN
TRPRNT
SEMANT
SELECT
RITER
RITEP
RITEI
PAGE
OUTA
OTHROAT
MESSAGE
LYPLOT
JTJ
IPJL
HISTORY
FSDUMP
CLIST

25. XTOI.

CALLED BY:

TXYZHXL
SEMANT
INITACQ
INIT
!

```

00001 1. MADEM
00002 2. QINTRY.
00003 2. RECCON
00004 3. TAPE6#
00005 3. OUTCI.
00006 3. ADUMP
00007 4. OUTCI.
00008 4. OUTCR.
00009 3. HALT
00010 4. HOLD
00011 5. ENTRYP
00012 6. MESSAGE
00013 7. TAPES#
00014 7. OUTCI.
00015 6. RITEI
00016 7. TAPES#
00017 7. OUTCI.
00018 6. RECER
00019 7. TAPES#
00020 7. OUTCI.
00021 7. OUTCR.
00022 6. TAPE6#
00023 6. OUTCI.
00024 6. ROUTER
00025 7. TAPES#
00026 7. OUTCI.
00027 7. OUTCR.
00028 6. ITRAP
00029 7. HALT (SEE LINE 00009)
00030 6. SECOND
00031 5. OUTBI.
00032 5. REWIND.
00033 5. EXITP
00034 6. SECOND
00035 6. MESSAGE (SEE LINE 00012)
00036 6. RITEI (SEE LINE 00015)
00037 6. RECER (SEE LINE 00018)
00038 6. TAPE6#
00039 6. OUTCI.
00040 6. ITRAP (SEE LINE 00028)
00041 6. ICHECK
00042 7. OUTCI.
00043 4. OUTCI.
00044 4. RITER
00045 5. TAPE6#
00046 5. OUTCI.
00047 4. RITEI (SEE LINE 00015)
00048 4. TRACE
00049 5. TRPRRT
00050 5. TRPRMT
00051 5. TRPRNT
00052 6. TAPE6#

```

```

00053          6. OUTCI.
00054      4. REGER      (SEE LINE 00018)
00055      4. ENTSTAT
00056          5. TAPE6#
00057          5. OUTCI.
00058      4. CLIST
00059          5. PAGE
00060              6. TAPE6#
00061              6. OUTCI.
00062          5. LNPLOT
00063              6. TAPE6#
00064              6. OUTCI.
00065          5. MESSAGE      (SEE LINE 00012)
00066          5. RITEP
00067              6. LCMLOC
00068              6. TAPE6#
00069              6. OUTCI.
00070          5. RITER      (SEE LINE 00044)
00071          5. RITEI      (SEE LINE 00015)
00072          5. TAPE6#
00073          5. OUTCI.
00074          5. OUTCR.
00075          5. CLIST2
00076              6. PAGE      (SEE LINE 00059)
00077              6. LNPLOT      (SEE LINE 00062)
00078              6. MESSAGE      (SEE LINE 00012)
00079              6. RITEI      (SEE LINE 00015)
00080              6. RITER      (SEE LINE 00044)
00081              6. TAPE6#
00082              6. OUTCI.
00083              6. OUTCR.
00084              6. RITEP      (SEE LINE 00066)
00085      4. PAGE      (SEE LINE 00059)
00086      4. ADUMP      (SEE LINE 00006)
00087      4. ISUMP
00088          5. OUTCI.
00089          5. OUTCR.
00090      4. ENOFIL.
00091      4. STOP.
00092      2. ENTRYP      (SEE LINE 00011)
00093      2. INPFI.
00094      2. DBGREAD
00095          3. OUTCI.
00096          3. INPCI.
00097          3. EOF
00098          3. DECODI.
00099      2. RECOVR
00100      2. GOTQER.
00101      2. FETCH
00102          3. ENTRYP      (SEE LINE 00011)
00103          3. INPFI.
00104          3. EXITP      (SEE LINE 00033)
00105      2. PAGE      (SEE LINE 00059)
00106      2. LCMLOC
00107      2. FSINIT

```

```

00108      3. ENTRYP      (SEE LINE 00011)
00109      3. EXITP      (SEE LINE 00033)
00110
00111      2. OTROAT
00112      3. ENTRYP      (SEE LINE 00011)
00113      3. GIMME
00114      4. ENTRYP      (SEE LINE 00011)
00115      4. HALT        (SEE LINE 00009)
00116      4. MESSAGE     (SEE LINE 00012)
00117      4. RITEI       (SEE LINE 00015)
00118      4. RITEP       (SEE LINE 00066)
00119      4. EXITP      (SEE LINE 00033)
00120      3. INPCI.
00121      3. TAPE6M
00122      3. OUTCI.
00123      3. RITEP      (SEE LINE 00066)
00124      3. ADDBLK
00125      4. ENTRYP      (SEE LINE 00011)
00126      4. EXITP      (SEE LINE 00033)
00127      3. INPFI.
00128      3. FINDBLK
00129      4. ENTRYP      (SEE LINE 00011)
00130      4. ROUTER      (SEE LINE 00024)
00131      4. MESSAGE     (SEE LINE 00012)
00132      4. RITEI       (SEE LINE 00015)
00133      4. CLIST      (SEE LINE 00058)
00134      4. EXITP      (SEE LINE 00033)
00135      3. HALT        (SEE LINE 00009)
00136      3. EXITP      (SEE LINE 00033)
00137      2. DISPAD
00138      3. OUTCI.
00139      3. DISPADS
00140      4. OUTCI.
00141      3. DISPF0B
00142      4. OUTCI.
00143      4. DISPFMF
00144      5. OUTCI.
00145      5. DISPFLT
00146      6. OUTCI.
00147      6. DISPPAY
00148      7. OUTCI.
00149      7. DISPPYB
00150      8. OUTCI.
00151      6. DISPADQ
00152      7. OUTCI.
00153      6. DISPADC
00154      7. OUTCI.
00155      6. DISPPRO
00156      7. OUTCI.
00157      3. DISPFLT     (SEE LINE 00144)
00158      3. DISPADC     (SEE LINE 00152)
00159      3. DISPPAF
00160      4. OUTCI.
00161      4. DISPPYB     (SEE LINE 00148)
00162      3. DISPPRO     (SEE LINE 00154)
00163      3. DISPADQ     (SEE LINE 00150)

```

```

00163      3. DISPADQ
00164      4. OUTCI.
00165      3. DISPACR
00166      4. OUTCI.
00167      4. DISPACL
00168      5. OUTCI.
00169      2. HALT      (SEE LINE 00009)
00170      2. INIT
00171      3. ENTRYP   (SEE LINE 00011)
00172      3. XTOI.
00173      3. GIMME    (SEE LINE 00112)
00174      3. RITEP    (SEE LINE 00066)
00175      3. SNAP
00176      4. ENTRYP   (SEE LINE 00011)
00177      4. EXITP    (SEE LINE 00033)
00178      3. EXITP    (SEE LINE 00033)
00179      2. LRKPRS
00180      3. SEMANT
00181      4. ENTRYP   (SEE LINE 00011)
00182      4. GOTDER.
00183      4. CODE01
00184      5. ENTRYP   (SEE LINE 00011)
00185      5. GIMME    (SEE LINE 00112)
00186      5. RITEP    (SEE LINE 00066)
00187      5. EXITP    (SEE LINE 00033)
00188      4. GIMME    (SEE LINE 00112)
00189      4. PACK
00190      5. ENTRYP   (SEE LINE 00011)
00191      5. LCMLOC
00192      5. PAGE     (SEE LINE 00059)
00193      5. MESSAGE  (SEE LINE 00012)
00194      5. RITEI    (SEE LINE 00015)
00195      5. TRACE    (SEE LINE 00048)
00196      5. ROUTER   (SEE LINE 00024)
00197      5. RITEP    (SEE LINE 00066)
00198      5. ISDUMP   (SEE LINE 00087)
00199      5. HALT     (SEE LINE 00009)
00200      5. EXITP    (SEE LINE 00033)
00201      4. RITEI    (SEE LINE 00015)
00202      4. RITEP    (SEE LINE 00066)
00203      4. LNPLT    (SEE LINE 00062)
00204      4. CODE03
00205      5. ENTRYP   (SEE LINE 00011)
00206      5. SRCHPL
00207      6. ENTRYP   (SEE LINE 00011)
00208      6. CREATE
00209      7. ENTRYP   (SEE LINE 00011)
00210      7. GIMME    (SEE LINE 00112)
00211      7. EXITP    (SEE LINE 00033)
00212      6. EXITP    (SEE LINE 00033)
00213      5. CREATE   (SEE LINE 00208)
00214      5. LOADPL
00215      6. ENTRYP   (SEE LINE 00011)
00216      6. EXITP    (SEE LINE 00033)
00217      5. RITEP    (SEE LINE 00066)

```

00218		
00219		
00220		
00221		
00222		
00223		
00224		
00225		
00226		
00227		
00228		
00229		
00230		
00231		
00232		
00233		
00234		
00235		
00236		
00237		
00238		
00239		
00240		
00241		
00242		
00243		
00244		
00245		
00246		
00247		
00248		
00249		
00250		
00251		
00252		
00253		
00254		
00255		
00256		
00257		
00258		
00259		
00260		
00261		
00262		
00263		
00264		
00265		
00266		
00267		
00268		
00269		
00270		
00271		
00272		
	5. LNPLOT	(SEE LINE 00062)
	5. ABQUEUE	
	6. ENTRYP	(SEE LINE 00011)
	6. GIMME	(SEE LINE 00112)
	6. ADDBLOK	(SEE LINE 00123)
	6. FINDBLK	(SEE LINE 00127)
	6. EXITP	(SEE LINE 00033)
	5. AOSASS	
	6. ENTRYP	(SEE LINE 00011)
	6. GIMME	(SEE LINE 00112)
	6. FINDBLK	(SEE LINE 00127)
	6. GOTOER.	
	6. DELAOD	
	7. ENTRYP	(SEE LINE 00011)
	7. MESSAGE	(SEE LINE 00012)
	7. TRACE	(SEE LINE 00048)
	7. RECEP	(SEE LINE 00018)
	7. GIMME	(SEE LINE 00112)
	7. SNAP	(SEE LINE 00175)
	7. LTRMRG	
	8. ENTRYP	(SEE LINE 00011)
	8. EXITP	(SEE LINE 00033)
	7. EXITP	(SEE LINE 00033)
	6. ADDBLOK	(SEE LINE 00123)
	6. EXITP	(SEE LINE 00033)
	5. EXITP	(SEE LINE 00033)
	4. GETHEX	
	5. ENTRYP	(SEE LINE 00011)
	5. HXDGTS	
	6. ENTRYP	(SEE LINE 00011)
	6. EXITP	(SEE LINE 00033)
	5. MESSAGE	(SEE LINE 00012)
	5. RITEI	(SEE LINE 00015)
	5. GIMME	(SEE LINE 00112)
	5. SCHATB	
	6. ENTRYP	(SEE LINE 00011)
	6. EXITP	(SEE LINE 00033)
	5. EXITP	(SEE LINE 00033)
	4. COOE05	
	5. ENTRYP	(SEE LINE 00011)
	5. SRCHPL	(SEE LINE 00206)
	5. FINDBLK	(SEE LINE 00127)
	5. GETHEX	(SEE LINE 00244)
	5. RITEI	(SEE LINE 00015)
	5. RITEP	(SEE LINE 00066)
	5. LNPLOT	(SEE LINE 00062)
	5. UOLLOAD	
	6. ENTRYP	(SEE LINE 00011)
	6. GIMME	(SEE LINE 00112)
	6. ADDBLOK	(SEE LINE 00123)
	6. EXITP	(SEE LINE 00033)
	5. ABQUEUE	(SEE LINE 00219)
	5. TGTLIST	
	6. ENTRYP	(SEE LINE 00011)
	6. FINDBLK	(SEE LINE 00127)

00273		
00274	6. GIMME	(SEE LINE 00112)
00275	6. ADDBLOK	(SEE LINE 00123)
00276	6. PELADD	
00277	7. ENTRYP	(SEE LINE 00011)
00278	7. GIMME	(SEE LINE 00112)
00279	7. PTRMRG	
00280	7. EXITP	(SEE LINE 00033)
00281	6. EXITP	(SEE LINE 00033)
00282	5. INITACQ	
00283	6. ENTRYP	(SEE LINE 00011)
00284	6. GIMME	(SEE LINE 00112)
00285	6. ADDBLOK	(SEE LINE 00123)
00286	6. ALOG.	
00287	6. MESSAGE	(SEE LINE 00012)
00288	6. RITEI	(SEE LINE 00015)
00289	6. XTOI.	
00290	6. NOWUCIT	
00291	7. ENTRYP	(SEE LINE 00011)
00292	7. HEXA00	
00293	8. ENTRYP	(SEE LINE 00011)
00294	8. IT0J.	
00295	8. EXITP	(SEE LINE 00033)
00296	7. HEXMJLT	
00297	8. ENTRYP	(SEE LINE 00011)
00298	8. IT0J.	
00299	8. EXITP	(SEE LINE 00033)
00300	7. GETHEX	(SEE LINE 00244)
00301	7. GIMME	(SEE LINE 00112)
00302	7. ADDRLOK	(SEE LINE 00123)
00303	7. FINDBLK	(SEE LINE 00127)
00304	7. DROPBLK	
00305	8. ENTRYP	(SEE LINE 00011)
00306	8. RELEASE	
00307	9. ENTRYP	(SEE LINE 00011)
00308	9. MESSAGE	(SEE LINE 00012)
00309	9. RITEI	(SEE LINE 00015)
00310	9. RITEP	(SEE LINE 00066)
00311	9. HALT	(SEE LINE 00009)
00312	9. GIMME	(SEE LINE 00112)
00313	9. EXITP	(SEE LINE 00033)
00314	8. EXITP	(SEE LINE 00033)
00315	7. RELEASE	(SEE LINE 00305)
00316	7. EXITP	(SEE LINE 00033)
00317	6. EXITP	(SEE LINE 00033)
00318	5. HISTORY	
00319	6. ENTRYP	(SEE LINE 00011)
00320	6. MESSAGE	(SEE LINE 00012)
00321	6. MASKER	
00322	6. TAPE64	
00323	6. OUTCI.	
00324	6. EXITP	(SEE LINE 00033)
00325	5. EXITP	(SEE LINE 00033)
00326	4. DMSDEC	
00327	5. ENTRYP	(SEE LINE 00011)
	5. EXITP	(SEE LINE 00033)

00328	
00329	
00330	
00331	
00332	
00333	
00334	
00335	
00336	
00337	
00338	
00339	
00340	
00341	
00342	
00343	
00344	
00345	
00346	
00347	
00348	
00349	
00350	
00351	
00352	
00353	
00354	
00355	
00356	
00357	
00358	
00359	
00360	
00361	
00362	
00363	
00364	
00365	
00366	
00367	
00368	
00369	
00370	
00371	
00372	
00373	
00374	
00375	
00376	
00377	
00378	
00379	
00380	
00381	
00382	
4. TLL2HX	
5. ENTRYP	(SEE LINE 00011)
5. SIN.	
5. TAN.	
5. TXY2HX	
6. ENTRYP	(SEE LINE 00011)
6. EXITP	(SEE LINE 00033)
5. CENTER	
6. ENTRYP	(SEE LINE 00011)
6. EXITP	(SEE LINE 00033)
5. IJ2HX	
6. HEXMLT	
7. HEXADD	(SEE LINE 00291)
7. ENTRYP	(SEE LINE 00011)
7. HXOGTS	(SEE LINE 00246)
7. MESSAGE	(SEE LINE 00012)
7. RITEI	(SEE LINE 00015)
7. EXITP	(SEE LINE 00033)
6. ENTRYP	(SEE LINE 00011)
6. ITOJ.	
6. EXITP	(SEE LINE 00033)
5. EXITP	(SEE LINE 00033)
4. INPCI.	
4. INPCR.	
4. TAPE6#	
4. OUTCI.	
4. OUTCR.	
4. ADDBLK	(SEE LINE 00123)
4. TTIME	
5. ENTRYP	(SEE LINE 00011)
5. EXITP	(SEE LINE 00033)
4. RITER	(SEE LINE 00044)
4. FINDBLK	(SEE LINE 00127)
4. CODE18	
5. GETHEX	(SEE LINE 00244)
5. GIMME	(SEE LINE 00112)
5. PACK	(SEE LINE 00189)
5. UNPACK	
6. ENTRYP	(SEE LINE 00011)
6. LCMLOC	
6. RITEP	(SEE LINE 00066)
6. ISDUMP	(SEE LINE 00087)
6. HALT	(SEE LINE 00009)
6. EXITP	(SEE LINE 00033)
5. UOLLOAD	(SEE LINE 00264)
5. RITEP	(SEE LINE 00066)
5. RITEI	(SEE LINE 00015)
5. LNPLT	(SEE LINE 00062)
5. TGTLIST	(SEE LINE 00270)
5. HISTORY	(SEE LINE 00317)
4. XTJI.	
4. SRCHPL	(SEE LINE 00206)
4. EXITP	(SEE LINE 00033)
3. TAPE6#	
3. OUTCI.	

00383	
00384	
00385	
00386	
00387	
00388	
00389	3. RDCCELL
00390	4. ISHIFT
00391	4. TAPE6#
00392	4. OUTCI.
00393	3. NXTSYM
00394	4. LEXAN
00395	5. CHRGEN
00396	6. CARD
00397	7. INPCI.
00398	7. EOF
00399	7. TAPE6#
00400	7. OUTCI.
00401	6. TAPE6#
00402	6. OUTCI.
00403	5. ISHIFT
00404	5. GOTOER.
00405	5. ADDCHR
00406	6. ISHIFT
00407	6. TAPE6#
00408	6. OUTCI.
00409	5. LOOKUP
00410	6. ISHIFT
00411	6. TAPE6#
00412	6. OUTCI.
00413	5. EXTSCN
00414	6. CHRGEN
00415	6. ITOJ.
00416	5. TAPE6#
00417	5. OUTCI.
00418	4. TAPE6#
00419	4. OUTCI.
00420	3. ERROR
00421	4. TAPE6#
00422	4. OUTCI.
00423	3. ISHIFT
00424	3. APCEL1
00425	4. ISHIFT
00426	4. TAPE6#
00427	4. OUTCI.
00428	3. APCEL2
00429	4. ISHIFT
00430	4. TAPE6#
00431	4. OUTCI.
00432	3. LACELL
00433	4. ISHIFT
00434	4. TAPE6#
00435	4. OUTCI.
00436	2. RELTST
00437	3. ENTRY# (SEE LINE 00011)
	3. RELEASE (SEE LINE 00305)
	3. EXIT# (SEE LINE 00033)
	2. DELADD (SEE LINE 00230)
	2. CONTROL
	3. ENTRY# (SEE LINE 00011)
	3. LTRF

(SEE LINE 00389)

00438	4. ENTRYP	(SEE LINE 00011)
00439	4. RELEASE	(SEE LINE 00305)
00440	4. LTRMRG	(SEE LINE 00237)
00441	4. EXITP	(SEE LINE 00033)
00442	3. SELECT	
00443	4. ENTRYP	(SEE LINE 00011)
00444	4. GOTDER.	
00445	4. MESSAGE	(SEE LINE 00012)
00446	4. RITEI	(SEE LINE 00015)
00447	4. HALT	(SEE LINE 00009)
00448	4. FELDEL	
00449	4. ASSIGN	
00450	4. ATTACK	
00451	4. COMMO	
00452	4. OOGFITE	
00453	4. ENGAGE	
00454	4. FLY	
00455	4. NAYBOR	
00456	4. PERCEPT	
00457	4. PLAN	
00458	5. ENTRYP	(SEE LINE 00011)
00459	5. GETPTRS	
00460	6. ENTRYP	(SEE LINE 00011)
00461	6. EXITP	(SEE LINE 00033)
00462	5. THTRPLN	
00463	6. ENTRYP	(SEE LINE 00011)
00464	6. CORBOUN	
00465	7. ENTRYP	(SEE LINE 00011)
00466	7. GIMME	(SEE LINE 00112)
00467	7. THM2PS	
00468	8. ENTRYP	(SEE LINE 00011)
00469	8. THX2XY	
00470	9. ENTRYP	(SEE LINE 00011)
00471	9. HXOGTS	(SEE LINE 00246)
00472	9. JJGGLE	
00473	10. ENTRYP	(SEE LINE 00011)
00474	10. EXITP	(SEE LINE 00033)
00475	9. EXITP	(SEE LINE 00033)
00476	8. EXITP	(SEE LINE 00033)
00477	7. TXY24XL	
00478	8. ENTRYP	(SEE LINE 00011)
00479	8. COS.	
00480	8. SIN.	
00481	8. XT01.	
00482	8. CENTER	(SEE LINE 00335)
00483	8. IJ2HX	(SEE LINE 00338)
00484	8. GETHEX	(SEE LINE 00244)
00485	8. EXITP	(SEE LINE 00033)
00486	7. HEXDIST	
00487	7. SIN.	
00488	7. COS.	
00489	7. LINEX	
00490	8. ENTRYP	(SEE LINE 00011)
00491	8. EXITP	(SEE LINE 00033)
00492	7. OPTPH	

00493	8. ENTRYP	(SEE LINE 00011)
00494	8. GIMMEI	(SEE LINE 00112)
00495	8. THW2PS	(SEE LINE 00467)
00496	8. HEXCHZ	
00497	9. ENTRYP	(SEE LINE 00011)
00498	9. TRACE	(SEE LINE 00048)
00499	9. MESSAGE	(SEE LINE 00012)
00500	9. RITEP	(SEE LINE 00066)
00501	9. CLOST	(SEE LINE 00058)
00502	9. HEXADD	(SEE LINE 00291)
00503	9. HEXINV	
00504	10. ENTRYP	(SEE LINE 00011)
00505	10. ITOJ.	
00506	10. EXITP	(SEE LINE 00033)
00507	9. GETHEX	(SEE LINE 00244)
00508	9. EXITP	(SEE LINE 00033)
00509	8. LINEX	(SEE LINE 00489)
00510	8. EXITP	(SEE LINE 00033)
00511	7. EXITP	(SEE LINE 00033)
00512	6. REVISE	
00513	7. ENTRYP	(SEE LINE 00011)
00514	7. GIMME	(SEE LINE 00112)
00515	7. ABVSCOR	
00516	8. ENTRYP	(SEE LINE 00011)
00517	8. CLOSCOR	
00518	9. ENTRYP	(SEE LINE 00011)
00519	9. HEXDIST	
00520	9. EXITP	(SEE LINE 00033)
00521	8. GIMMEI	(SEE LINE 00112)
00522	8. ADJBLK	(SEE LINE 00123)
00523	8. EXITP	(SEE LINE 00033)
00524	7. RELEASE	(SEE LINE 00305)
00525	7. EXITP	(SEE LINE 00033)
00526	6. FINDBLK	(SEE LINE 00127)
00527	6. KOMPARE	
00528	7. ENTRYP	(SEE LINE 00011)
00529	7. UNPACK	(SEE LINE 00365)
00530	7. PACK	(SEE LINE 00189)
00531	7. EXITP	(SEE LINE 00033)
00532	6. CANNOTGT	
00533	7. ENTRYP	(SEE LINE 00011)
00534	7. FINDBLK	(SEE LINE 00127)
00535	7. CLOSCOR	(SEE LINE 00517)
00536	7. JGESJIT	
00537	8. ENTRYP	(SEE LINE 00011)
00538	8. THX2XY	(SEE LINE 00469)
00539	8. ATANZ.	
00540	8. EXITP	(SEE LINE 00033)
00541	7. FORMTGT	
00542	8. ENTRYP	(SEE LINE 00011)
00543	8. GIMMEI	(SEE LINE 00112)
00544	8. FINDBLK	(SEE LINE 00127)
00545	8. FINDFLT	
00546	9. ENTRYP	(SEE LINE 00011)
00547	9. HEXOIST	

00548		9. FINOBLK	(SEE LINE 00127)
00549		9. KOMPARE	(SEE LINE 00527)
00550		9. GIMME	(SEE LINE 00112)
00551		9. ADDBLOK	(SEE LINE 00123)
00552		9. EXITP	(SEE LINE 00033)
00553		8. RELEASE	(SEE LINE 00305)
00554		8. EXITP	(SEE LINE 00033)
00555		7. GIMME	(SEE LINE 00112)
00556		7. ADDBLOK	(SEE LINE 00123)
00557		7. TGTGONE	
00558		8. ENTRYP	(SEE LINE 00011)
00559		8. PTREE	
00560		9. ENTRYP	(SEE LINE 00011)
00561		9. RELEASE	(SEE LINE 00305)
00562		9. ISHIFT	
00563		9. EXITP	(SEE LINE 00033)
00564		8. GIMME	(SEE LINE 00112)
00565		8. EXITP	(SEE LINE 00033)
00566		7. PTREE	(SEE LINE 00559)
00567		7. RANDOM.	
00568		7. PELADD	(SEE LINE 00275)
00569		7. EXITP	(SEE LINE 00033)
00570		6. ITRAP	(SEE LINE 00028)
00571		6. AVAILBL	
00572		7. ENTRYP	(SEE LINE 00011)
00573		7. PELADD	(SEE LINE 00275)
00574		7. RELEASE	(SEE LINE 00305)
00575		7. EXITP	(SEE LINE 00033)
00576		6. SCHEDUL	
00577		7. ENTRYP	(SEE LINE 00011)
00578		7. UNPACK	(SEE LINE 00365)
00579		7. RANDOM.	
00580		7. RENDEUV	
00581		8. ENTRYP	(SEE LINE 00011)
00582		8. TXK2XY	(SEE LINE 00469)
00583		8. TXY24XL	(SEE LINE 00477)
00584		8. EXITP	(SEE LINE 00033)
00585		7. HEXADD	(SEE LINE 00291)
00586		7. GETHEX	(SEE LINE 00244)
00587		7. HEXDIST	
00588		7. ACFRAG	
00589		8. ENTRYP	(SEE LINE 00011)
00590		8. GIMME	(SEE LINE 00112)
00591		8. CRFLTYL	
00592		9. ENTRYP	(SEE LINE 00011)
00593		9. CREATE	(SEE LINE 00208)
00594		9. GIMME	(SEE LINE 00112)
00595		9. ADDBLOK	(SEE LINE 00123)
00596		9. HISTORY	(SEE LINE 00317)
00597		9. EXITP	(SEE LINE 00033)
00598		8. ADDBLOK	(SEE LINE 00123)
00599		8. OPTPTH	(SEE LINE 00492)
00600		8. RELEASE	(SEE LINE 00305)
00601		8. FLTGROM	
00602		9. ENTRYP	(SEE LINE 00011)

00603		9. MEXCHZ	(SEE LINE 00496)
00604		9. TH2PS	(SEE LINE 00467)
00605		9. ATAN2.	
00606		9. EXITP	(SEE LINE 00033)
00607		8. HEXDIST	
00608		8. DELADD	(SEE LINE 00230)
00609		8. EXITP	(SEE LINE 00033)
00610		7. EXITP	(SEE LINE 00033)
00611		6. DELADD	(SEE LINE 00230)
00612		6. PLANOUT	
00613		7. ENTRYP	(SEE LINE 00011)
00614		7. OUTCI.	
00615		7. EXITP	(SEE LINE 00033)
00616		6. RLRAID	
00617		7. ENTRYP	(SEE LINE 00011)
00618		7. RLWAVE	
00619		8. ENTRYP	(SEE LINE 00011)
00620		8. RLGTYP	
00621		9. ENTRYP	(SEE LINE 00011)
00622		9. RELEASE	(SEE LINE 00305)
00623		9. RLGTAK	
00624		10. ENTRYP	(SEE LINE 00011)
00625		10. RLFMAKT	
00626		11. ENTRYP	(SEE LINE 00011)
00627		11. RELEASE	(SEE LINE 00305)
00628		11. EXITP	(SEE LINE 00033)
00629		10. RELEASE	(SEE LINE 00305)
00630		10. EXITP	(SEE LINE 00033)
00631		9. EXITP	(SEE LINE 00033)
00632		8. RELEASE	(SEE LINE 00305)
00633		8. EXITP	(SEE LINE 00033)
00634		7. RLCORO	
00635		8. ENTRYP	(SEE LINE 00011)
00636		8. RLABJ8	
00637		9. ENTRYP	(SEE LINE 00011)
00638		9. RELEASE	(SEE LINE 00305)
00639		9. EXITP	(SEE LINE 00033)
00640		8. RELEASE	(SEE LINE 00305)
00641		8. EXITP	(SEE LINE 00033)
00642		7. RELEASE	(SEE LINE 00305)
00643		7. EXITP	(SEE LINE 00033)
00644		6. EXITP	(SEE LINE 00033)
00645		5. EXITP	(SEE LINE 00033)
00646		4. PONDER	
00647		4. TOWER	
00648		4. UMPIRE	
00649		4. TAPE6#	
00650		4. OUTCI.	
00651		4. OUTER.	
00652		4. EXITP	(SEE LINE 00033)
00653	3. SNAP		(SEE LINE 00175)
00654	3. UNSNAP		
00655	4. ENTRYP		(SEE LINE 00011)
00656	4. EXITP		(SEE LINE 00033)
00657	3. RELEASE		(SEE LINE 00305)

00658 3. SECOND
00659 3. HLTBNT
00660 4. ENTRYP (SEE LINE 00011)
00661 4. SECOND
00662 4. HALT (SEE LINE 00009)
00663 4. EXITP (SEE LINE 00033)
00664 3. EXITP (SEE LINE 00033)
00665 2. EXITP (SEE LYVE 00033)
00666 2. END.

2. Main Processor

PAGE 1

LIST OF SUBROUTINES - WADEN

1. ABSEE
2. ABVSCOR
3. ABZCRC
4. ACCEPT
5. ACFRAG
6. ADDBLOK
7. ADUMP
8. AIRTHNK
9. ALLOBAT
10. ALLOFU
11. ALLOPAT
12. AMMOCHK
13. ASSIGN
14. ATKASES
15. ATTACK
16. AUTOPRI
17. AVAILBL
18. AZILIN
19. BADMOVE
20. BATCEAS
21. BATTCOV
22. BATTOUT
23. BLKDAT
24. BNCMOPR
25. BNCONHO
26. BNCONLS
27. BNCONTC
28. BNLALLE
29. BNNOTRD
30. BNNWTRK
31. BNPNBB
32. BNPNBD
33. BNPNDA
34. BNPNEP
35. BNPNFA
36. BNPNFD
37. BNPNSS
38. BNRECOV
39. BOCTINK
40. BTNASIN
41. BTN2CRC
42. BTRYTHK
43. BYALCOV
44. BYCMOPR
45. BYCONHO
46. BYCONLS
47. BYCONTC
48. BYENOPS
49. BYHEDUP
50. BYNOTRD
51. BYNWTRK
52. BYPASUP

53. BYPNER
54. BYPONFO
55. BYPONRL
56. BYPONRS
57. BYPONTM
58. BYTKCHK
59. BYUPDAT
60. CANCALO
61. CANDTGT
62. CENTER
63. CFLYCRC
64. CHKCOV
65. CHKLAST
66. CLIST
67. CLIST2
68. CLOSCOR
69. CNACTTK
70. COMMAND
71. COMMO
72. CONTROL
73. CORBOUN
74. COVAPLY
75. CRCDIES
76. CRCEVNT
77. CRCKTL
78. CRCLJSS
79. CRCSEE
80. CRCTHMK
81. CRCTRAK
82. CRC2INT
83. CREATE
84. CRFLTML
85. DBGREAD
86. DECRALO
87. DELADD
88. DESTROY
89. DETECT
90. DBTSHX
91. DILOUT
92. DISPABQ
93. DISPACD
94. DISPACL
95. DISPACR
96. DISPADS
97. DISPAGD
98. DISPDAT
99. DISPFOB
100. DISPFLT
101. DISPFNF
102. DISPPAF
103. DISPPAY
104. DISPPRO
105. DISPPYB
106. OLYACT
107. OOGFITE

108. DOGTHNK
109. DROPBLK
110. DROPP0S
111. DROPPS2
112. ENGAGE
113. ENTRYP
114. ENTSTAT
115. EOF
116. EXITP
117. FELDEL
118. FETCH
119. FILERUP
120. FINOBLK
121. FINOFLT
122. FINDIT
123. FIRECHK
124. FLITE
125. FLTGEOM
126. FLTWYPE
127. FLY
128. FLYSEE
129. FORMTGT
130. FSDUMP
131. FSINIT
132. FUELCHK
133. GETHEX
134. GETPTRS
135. GIMME
136. GNLDJOK
137. GOGETEM
138. GOTOAB
139. HALT
140. HANZPT
141. HEXADD
142. HEXCHZ
143. HEXDIST
144. HEXINV
145. HEXMLT
146. HEXMOVE
147. HEXMULT
148. HISTORY
149. HLTPNT
150. HOLD
151. HXDGTS
152. HXMLT2
153. ICHECK
154. IJ2HX
155. INIT
156. INITACO
157. INRANGE
158. INSECT
159. INSERT
160. INTASIN
161. INTFIND
162. INTRFLY

163. INT2CRC
164. IPJL
165. ISDUMP
166. ISHIFT
167. ITRAP
168. JGESUIT
169. JTJ
170. JUGGLE
171. KILFLIT
172. KOMPARE
173. LCML0C
174. LINEX
175. LNPL0T
176. LOADPL
177. LOSRADR
178. LRKPRS
179. LTRER
180. LTRMRG
181. MADE4
182. MASKER
183. MESSAGE
184. MESBILD
185. NAYBOR
186. NEWMOVE
187. NEWPERC
188. NOWUCIT
189. NUKBLND
190. OPTPTH
191. OTHROAT
192. OUTA
193. OUTPTRS
194. PACK
195. PAGE
196. PATDEC
197. PELADD
198. PERCEPT
199. PLAN
200. PLANOUT
201. PONDER
202. PREPAFU
203. PRIORITY
204. PTRONER
205. PTRAND
206. PTREE
207. PTRMRG
208. READIL
209. RECCON
210. REGER
211. RECOVR
212. REDEARR
213. RELEASE
214. RELIST
215. RELOAD
216. RELOCAT
217. RELSTLL

273. UNLINK
274. UNPACK
275. UNSNAP
276. UNSTAT
277. UOLLQAO
278. *WIPEOUT
279. *THDRAW
280. XPAA
281. XPD
282. XPK
283. XSHIFT
284. XY2HX
285. YANK

PAGE 6

LIST OF FORTRAN LIBRARY ROUTINES - MADM

1. ALOG.
2. ASIN.
3. ATAN2.
4. COS.
5. DECOOI.
6. ENDFIL.
7. END.
8. GOTOER.
9. INPBI.
10. INPCI.
11. INPFI.
12. ITOJ.
13. OUTBI.
14. OUTCI.
15. OUTCR.
16. QINTRY.
17. RANDOM.
18. REWIND.
19. SIN.
20. SQRT.
21. STOP.
22. TAN.
23. TAPE6#
24. XTOI.

218. RENDEUV
219. RESUPLY
220. REVISE
221. RITEI
222. RITEP
223. RITER
224. RLABDB
225. RLCORO
226. RLFMAKT
227. RLRATO
228. RLGTAK
229. RLGTYP
230. RLWAVE
231. RONDSEE
232. ROUTER
233. SAMATON
234. SAMPRCM
235. SAMSEE
236. SAMWYPE
237. SCHEDUL
238. SCHATB
239. SDIGEST
240. SECONO
241. SEEKENG
242. SEEXP
243. SEEKTAC
244. SEEKTFU
245. SELECT
246. SETASSN
247. SHRKILL
248. SHUFFLE
249. SKSBTRK
250. SNAP
251. SSSL
252. STATPAK
253. STICK
254. TERMACQ
255. TFLYCRC
256. TGTGONE
257. TGTHEX
258. THW2DS
259. THTRPLN
260. THX2XY
261. TH2HX
262. TOADIL
263. TOWER
264. TRACE
265. TRKCHK
266. TRPRWT
267. TRPRNT
268. TRPRRT
269. TRYSHOT
270. TXY2HX
271. TXY24XL
272. UMPIRE

SUBROUTINE REFERENCE LIST - MADE4

PAGE 9

1. ABSEE	CALLS: UNPACK DELOAD RELEASE TAPE6# OUTCI.	CALLED BY: PERCEPT
2. ABVSCOR	CALLS: ENTRYP CLOSCOR GIMME ADDBLOK EXITP	CALLED BY: REVISE
3. ABZCRC	CALLS: UNPACK RELEASE FINOBLK GIMME ADDBLOK TAPE6# OUTCI. DROPBK	CALLED BY: CRCTHNK
4. ACCEPT	CALLS: UNPACK SEEKP GIMME STOP. STICK DETECT DELOAD PACK TAPE6# OUTCI. BYALCOV RELEASE BYHEDUP OUTA	CALLED BY: BYCMDPR BYCMDPR
5. ACFRAG	CALLS: GIMME UNPACK CRFLTM. ADDBLOK OPTPTH RELEASE PACK FLTGEOM HEXDIST DELOAD	CALLED BY: SCHEDUL

6. A00BLOK	CALLS: ENTRYP EXITP	CALLED BY: UOLLOAD REDEBRF NOWUCIT NEWMOVE INT2CRC INITACQ GOGETEM FLYSEE FINDIT FINOFLT CRFLTML CANOTGT BADMOVE ATKASES ACFRAG ABVSCOR AB2CRC
7. ADUMP	CALLS: OUTCI. OUTCR.	CALLED BY: HALT RECCON
8. AIRTHNK	CALLS: HEXDIST MESBILO DELAOD TGTHX TAPE6# OUTCI. RANDOM.	CALLED BY: TFLYCRC
9. ALLOBAT	CALLS: PRIORTY MESBILO GIMME PACK TAPE6# OUTCI. OUTA DELAOD YANK STICK TOADIL	CALLED BY: SEEKTAC SEEKENG BOCTINK
10. ALLOFU	CALLS: GIMME STICK DELAOD TAPE6# OUTCI.	CALLED BY: SEEKTFU BATTCOV
11. ALLOPAT	CALLS: GIMME	CALLED BY: SEEKTFU

	STICK DELAOD TAPE6# OUTCI.	BATTCOV
12. AMMOCHK	CALLS: MESBILO DELAOD GOTOER. BYNOTRO TAPE6# OUTCI.	CALLED BY: ENGAGE
13. ASSIGN	CALLS: GETPTRS INTASIN	CALLED BY: SELECT
14. ATKASES	CALLS: GIMME DELAOD DESTROY TAPE6# OUTCI. ADDBLOK THH2PS ATAN2.	CALLED BY: CFLYCRC
15. ATTACK	CALLS: GETPTRS UNSTAT SHRKILI FINDBLK XPK RANDOM. NUKBLND DROPLK HISTORY DELAOD STATPAK	CALLED BY: SELECT
16. AUTOPRI		CALLED BY: SAMATON PREPAFU BYCONHO
17. AVAILBL	CALLS: PELAOD RELEASE	CALLED BY: THTRPLN
18. AZILIM	CALLS: SIN. COS. INSECT SQRT.	CALLED BY: INRANGE

19. BAOMOVE	CALLS: FINDBLK GIMME ADDBLK HISTORY UNPACK RELEASE DROPBLK TGTHEX HEXDIST MESBILD DELAOD	CALLED BY: CRCTRK
20. BATCEAS	CALLS: CANCEL GIMME UNPACK SEEKFI STICK RELEASE	CALLED BY: TRYSHOT HANZPT BYCONTC BYPONFO BYNOTRO BYCONLS BYCMOPR BTRYTNK
21. BATTCOV	CALLS: TRKCHK ALLOFU ALLOPAT	CALLED BY: BYALCOV BYCONTC BYPONER BYNWTRK BYHEDUP BYCONHO
22. BATTOUT	CALLS: YANK BNLALLE UNPACK CHKLAST GIMME RELEASE SSSL	CALLED BY: BYUPDAT BNPONBO
23. BLKDAT		
24. BNCMDPR	CALLS: GOTDER. ACCEPT MESBILD DELAOD SEEKP DROPPS DILOUT	CALLED BY: BOCTINK
25. BNCONHO	CALLS: TAPE6# OUTCI.	CALLED BY: BNCONTC

GIMME
 MESBILD
 PRIORITY
 PACK
 DELADD
 YANK
 STICK
 RELEASE
 DROPPOS
 SEEKTAC
 DLYACT

26. BNCONLS

CALLS:

DROPPOS
 OILOUT
 MESBILD
 DELADD

CALLED BY:

SDIGEST

27. BNCONTC

CALLS:

CHKCOV
 DROPPOS
 BNLALLE
 SETASSN
 SEEKTAC
 GIMME
 DLYACT
 CHKLAST
 BNCONHD
 BNRECOV
 TAPES#
 OUTCI.
 YANK
 STICK
 TOADIL
 OILOUT

CALLED BY:

SDIGEST
 BNPONDA

28. BNLALLE

CALLS:

MESBILD
 DELADD
 YANK
 HANOZPT
 OILOUT
 STICK

CALLED BY:

TRYSHOT
 SDIGEST
 TRKCHK
 SAMPRM
 BYCONTC
 BNCONTC
 BYNWTRK
 BYENDPS
 BYCONLS
 BTRYTNK
 BOCTINK
 BNPONE?
 BNPONDA
 BNNWTRK
 BATTOUT

29. BNNOT#0

CALLS:

TAPE6#

CALLED BY:

BYUPDAT

OUTCI.
RELEASE
DILOUT

BNPONBD

30. BNNWTRK

CALLS:

CHKCOV
BNLALLE
SETASSN
SEEKTAC
GIMME
DLYACT

CALLED BY:

SOIGEST

31. BNPNB8

CALLS:

FILERUP
UNPACK
INRANGE
GIMME
DELADD
SETASSN
SEEKENG
PRIORITY
RELOCAT
STICK
GOTOER.
DLYACT
TAPE6#
OUTCI.

CALLED BY:

BNPNSS

32. BNPNAD

CALLS:

TAPE6#
OUTCI.
BNNOTRO
BATTOUT
COVAPLY
RELEASE

CALLED BY:

BOCTINK

33. BNPNDA

CALLS:

YANK
UNPACK
GOTDER.
SETASSN
SEEKTAC
CHKCOV
BNLALLE
GIMME
DLYACT
BNCONTC
SEEKENG
RELEASE

CALLED BY:

BOCTINK

34. BNPNEP

CALLS:

GOTDER.
SKSBRK
DROPPS
BNLALLE

CALLED BY:

BOCTINK

UNPACK
 CHKLAST
 SEEKTAC
 GIMME
 PACK
 DLYACT
 SEEKENG
 MESBILD
 DELADD
 YANK
 STICK
 READIL

35. BNPONFA	CALLS: SEEKP TAPE6# OUTCI. BNRECOV	CALLED BY: BOCTINK
36. BNPONFD	CALLS: SEEKP TAPE6# OUTCI. RELEASE YANK DROPPOS DILOUT MESBILD DELAOD	CALLED BY: BOCTINK
37. BNPONSS	CALLS: UNPACK GOTOER. BNPONBR SEEKENG GIMME DLYACT BYUPOAT	CALLED BY: BOCTINK
38. BNRECOV	CALLS: UNPACK WITHDRAW PACK	CALLED BY: BNCONTC BNPONFA
39. BOCTINK	CALLS: BNPONSQ BNPONEP BNCMDPR BNPONFA BNPONFD RELEASE SDIGEST BNPONDA TAPE6# OUTCI.	CALLED BY: PONDER

DROPOS
BNLALLE
MESBILO
DELAOD
DILOUT
BNPONBD
SAMATON
ALLOBAT

40. BTNASTN

CALLS:

UNPACK
FINDBLK
PACK
HEXDIST
RELEASE
DROPBK
MESBILO
DELAOD
TAPE6#
OUTCI.

CALLED BY: INTASIN

41. BTN2C3C

CALLS:

UNPACK
RELEASE
TAPE6#
OUTCI.
CRCLOSS
CRCKIL
CRCTRK

CALLED BY: CRCTHMK

42. BTRYTK

CALLS:

BYCMOPR
SDIGEST
SEEKP
BYPONTM
BYPONER
TAPE6#
OUTCI.
YANK
BATCEAS
BNLALLE
MESBILO
DELAOD
BYPONFD
SAMATON
BYPONRL
BYPONRS

CALLED BY: PONDOR

43. BYALCOV

CALLS:

UNPACK
PACK
GIMME
SLL
CANCALO
SEEKTFU

CALLED BY: BYCONTC
BYCMOPR
ACCEPT

RELEASE
PATDEC
BATTCOV
DLYACT
TAPE6#
OUTCI.

44. BYCMOPR	CALLS: ACCEPT SEEKP TAPE6# OUTCI. BATCEAS DILOUT BYALCOV BYHEDUP RELEASE	CALLED BY: BTRYTNK
45. BYCONHO	CALLS: TAPE6# OUTCI. DELADD AUTOPRI STICK RELOCAT BATTCOV GIMME DLYACT	CALLED BY: BYCONTC
46. BYCONLS	CALLS: BATCEAS BNLALLE MESBILD DELADD DILOUT	CALLED BY: SOIGEST
47. BYCONTC	CALLS: MESBILD DELADD INRANGE BATCEAS BNLALLE PREPAFU BATTCOV GIMME DLYACT BYCONHO UNPACK BYALCOV STICK RELOCAT YANK TOADIL	CALLED BY: SOIGEST BYPONTM
48. BYENDPS	CALLS:	CALLED BY:

SEEKP
UNPACK
YANK
RELEASE
GETPTRS
CRCLOSS
BNLALLE

SAMWPE
SAMPRCM
BYPONFO

49. BYHEDUP

CALLS:

SEEKP
UNPACK
DELADD
RELOCAT
BATTMOV
MESBILD
RELEASE

CALLED BY:

BYCMOPR
ACCEPT

50. BYNOTRD

CALLS:

UNPACK
BATCEAS
OILOUT
TAPE6#
OUTCI.

CALLED BY:

AMMOCHK

51. BYNWTRK

CALLS:

MESBILO
DELADD
BNLALLE
INRANGE
PREPAF(;
BATTMOV
GIMME
OLYACT
STICK
RELOCAT

CALLED BY:

SOIGEST

52. BYPASJP

CALLS:

SEEKP
UNPACK
GIMME
PACK
NEWPERC
GETPTRS
CRCTRK

CALLED BY:

SAMPRCM

53. BYPONER

CALLS:

UNPACK
CANCALO
SEEKTFU
BATTMOV
OILOUT
PACK
DELADD
PTPONER
MESBILO

CALLED BY:

BTRYTNK

RELEASE

54. BYPONFD	CALLS: SEEKP YANK RELEASE BATCEAS DILOUT BYENOPS MESBILD DELADD	CALLED BY: BTRYTNK
55. BYPONRL	CALLS: UNPACK GOTOER. FILERUP INRANGE PREPAFU MESBILD DELAOD SEEKTFU TAPE6# OUTCI.	CALLED BY: BTRYTNK
56. BYPONRS	CALLS: UNPACK RELOAD TAPE6# OUTCI. RELEASE	CALLED BY: BTRYTNK
57. BYPONTM	CALLS: GIMME UNPACK PACK TAPE6# OUTCI. BYCONTC TRYSHOT DELADD RELEASE	CALLED BY: BTRYTNK
58. BYTKCHK	CALLS: SQRT. ATAN2. LOSRADR DELADD YANK STICK TOADIL	CALLED BY: SAMPRCM
59. BYUPOAT	CALLS: UNPACK BNNOTRO BATTOUT	CALLED BY: BYPONSS

COVAPLY
GIMME
PACK
SSLL
DECRALO
SEEKTAC
OLYACT
RELEASE

60. CANCALO	CALLS: YANK RELEASEF TAPE6# OUTCI. READIL	CALLED BY: PTPNER BYALCOV PATDEC BYPNER BATCEAS
61. CANDTGT	CALLS: ENTRYP FINOBLK CLOSCOR JGESUIT FORMTGT GIMME ADDBLCK TGTGONE PTREE RANDOM. PELADD EXITP	CALLED BY: THTRPLV
62. CENTER	CALLS: ENTRYP EXITP	CALLED BY: XYZHX TXYZHXL
63. CFLYCRC	CALLS: CRCSEE UNSTAT FLYSEE ATXASES CRC2INT RONDSEF GNOLOOK STATPAK	CALLED BY: PERCEPT
64. CHKCOV	CALLS: UNPACK INRANGE GIMME TAPE6# OUTCI. OUTA DELAOD	CALLED BY: BYCONTC BNPONA BNWTRK
65. CHKLAST	CALLS: DROPPOS	CALLED BY: BYCONTC

	PRIORITY RELOCAT DELAOD	BNPONEP BATTOUT
66. CLIST	CALLS: PAGE LNPLOT MESSAGE RITEP RITER RITEI TAPE6# OUTCI. OUTCR. CLIST2	CALLED BY: HALT UNSTAT TGTHEX TRKCHK SAMPRCM NAYBOR HEXMOVE HEXCHZ FSOUMP FLTWYPE FINDBLK CRCDIES INTASIN
67. CLIST2	CALLS: PAGE LNPLOT MESSAGE RITEI RITER TAPE6# OUTCI. OUTCR. RITEP	CALLED BY: CLIST
68. CLOSCOR	CALLS: ENTRYP HEXOIST EXITP	CALLED BY: CANDTGT ABVSCOR
69. CNACTTK	CALLS: YANK RELEASE	CALLED BY: DILOUT
70. COMMAND	CALLS: UNPACK GOTOER. DELAOD TAPE6# OUTCI. MESBILD FLTWYPE THM2PS ATAN2.	CALLED BY: FLITE
71. COMMO	CALLS: UNPACK DELAOD RELEASE PACK	CALLED BY: SELECT

72. CONTROL	CALLS: ENTRYP LTREE SELECT SNAP UNSNAP RELEASE SECOND HLTPT EXITP	CALLED BY: MADEM
73. CORBOUN	CALLS: ENTRYP GIMME THM2PS TXY2HXL HEXD1ST SIN. COS. LINEX OPTPTH EXITP	CALLED BY: THTRPLN
74. COVAPLY	CALLS: SEEKTAC GIMME DLYACT RELEASE	CALLED BY: BYUPDAT BNPONBO
75. CRCDIES	CALLS: UNPACK RELEASE DELADD FINDBLK TRACE MESSAGE RITEI CLIST DROPBLK	CALLED BY: DESTROY
76. CRCEVNT	CALLS: GIMME TAPE6# OUTCI. DELADD RELEASE FINDBLK DETECT UNPACK HEXD1ST LOSRAOR COS. RANOOM. CRCTRAK	CALLED BY: CRCSEE

77. CRCKIL	CALLS: FINOBLK UNPACK RELEASE DROPBLK TAPE6# OUTCI. DELADD	CALLED BY: INT2CRC CRCTRAK CRCTHNK BTN2CRC
78. CRCLOSS	CALLS: FINOBLK UNPACK DROPBLK DELADD	CALLED BY: SAMPRCM CRCTHNK BYENDPS BTN2CRC
79. CRCSEE	CALLS: RELEASE GIMME UNPACK DELADD CRCEVNT	CALLED BY: CFLYCRC
80. CRCTHNK	CALLS: UNPACK RELEASE CRCKIL CRCTRAK CRCLOSS AB2CRC INT2CRC BTN2CRC	CALLED BY: TFLYCRC
81. CRCTRAK	CALLS: FINOBLK UNPACK MESBILD DELADD TAPE6# OUTCI. CRCKIL BADMOVE DROPBLK RANDOM. NEWMOVE	CALLED BY: SAMPRCM CRCTHNK CRCEVNT BYPASUP BTN2CRC
82. CRC2INT	CALLS: UNPACK TAPE6# OUTCI. MESBILD DELADD OPTPTH RELEASE THH2PS	CALLED BY: CFLYCRC

	ATAN2.	
83. CREATE	CALLS: ENTRYP GIMME EXITP	CALLED BY: CRFLTML
84. CRFLTML	CALLS: CREATE GIMME UNPACK PACK ADDBLOK HISTORY	CALLED BY: GOGETEM ACFRAG
85. OBGREAD	CALLS: OUTCI. INPCI. EOF DECOOI.	CALLED BY: MADEM
86. DECRAIO	CALLS: MESBILO DELADD TAPE6# OUTCI.	CALLED BY: #THORAW BYUPOAT
87. DELAD9	CALLS: ENTRYP MESSAGE TRACE RECER GIMME SNAP LTRMRG EXITP	CALLED BY: UMPIRE TOWER THTRPLN SHRKILL SETASSN SEEKTAC SEEKENG SOIGEST SAMWYPE SAMSEE SAMATON RESUPLY RELOAD READIL PTPNER PREPAFU NUKBLNO NEWPERC NEWMOVE BYTKCHK BYCONTC BNLALLE MAYBOR XILFLIT INTRFLY INTZCRC INTFINO

CRCZINT
GOGETEM
GNDLOOK
FUELCHK
FLYSEE
FLY
FLITE
FILERUP
ENGAGE
DROPPS2
DROPPS
DOGTHNK
DOGFITE
DESTROY
DECRALO
CRCTRAK
CRCSEE
CRCLOSS
CRCXIL
CRCEVNT
CRCDIES
COMMO
COMMAND
CHKLAST
CHKCOV
BYPONTH
BYPONRL
BYPONFD
BYPONER
BYNWTRK
BYHEDUP
BYCONLS
BYCONHO
BTRYTNK
BTNASIN
BOCTINK
BNPONFD
BNPONEP
BNPONBB
BNCONLS
BNCONHO
BNCMDPR
BADMOVE
INTASIN
ATTACK
ATKASES
AMMOCHK
ALLOPAT
ALLOFU
ALLOBAT
AIRTHNK
ACFRAG
ACCEPT
ASSEE
MADEM

88. DESTROY	CALLS: TAPE6# OUTCI. GETPTRS TERMACQ KILFLIT GIMME DELADD SAMWYPE CRCOIES UNLINK RELEASE UNSTAT	CALLED BY: UMPIRE TOWER SHRKILL REDEBRF FLY DOGFITE ATKASES
89. DETECT	CALLS: HEXDIST UNPACK THH2PS ATAN2. LOSRADR COS. RANDOM. TAPE6# OUTCI.	CALLED BY: SAMPRCM INT2CRC FLYSEE CRCEVNT ACCEPT
90. DGTSMX	CALLS: ITOJ.	
91. DILOUT	CALLS: CNACTTK RELEASE UNPACK YANK TOADIL	CALLED BY: SAMATON PTPNER HANDZPT BNLALLE BNCONTC BYPONFD BYPONER BYNOTRD BYCONLS BYCMOPR BOCTINK BNPONFD BNNOTRD BNCONLS BNCMOPR
92. DISPABQ	CALLS: OUTCI.	CALLED BY: DISPDAT
93. DISPACD	CALLS: OUTCI.	CALLED BY: DISPFLT DISPOAT
94. DISPACL	CALLS: OUTCI.	CALLED BY: DISPACR

95. DISPACR	CALLS: OUTCI. DISPACL	CALLED BY: DISPOAT
96. DISPADS	CALLS: OUTCI.	CALLED BY: DISPOAT
97. DISPAQD	CALLS: OUTCI.	CALLED BY: DISPFLT DISPOAT
98. DISPOAT	CALLS: OUTCI. DISPADS DISPFDR DISPFLT DISPACD DISPPAF DISPPRO DISPAQD DISPABQ DISPACR	CALLED BY: MADEM
99. DISPFDB	CALLS: OUTCI. DISPFMF	CALLED BY: DISPOAT
100. DISPFLT	CALLS: OUTCI. DISPPAY DISPAQD DISPACD DISPPRO	CALLED BY: DISPFMF DISPOAT
101. DISPFMF	CALLS: OUTCI. DISFFLT	CALLED BY: DISPFDB
102. DISPPAF	CALLS: OUTCI. DISPPYB	CALLED BY: DISPOAT
103. DISPPAY	CALLS: OUTCI. DISPPYB	CALLED BY: DISPFLT
104. DISPPRO	CALLS: OUTCI.	CALLED BY: DISPFLT DISPOAT
105. DISPPYB	CALLS: OUTCI.	CALLED BY: DISPPAY DISPPAF

106. DLYACT

CALLS:
UNPACK
PACK
STICK

CALLED BY:
WTHORAW
SOIGEST
BYALCOV
BYCONTC
BNCONTC
DROPPS2
DROPPS
COVAPLY
BYUPDAT
BYNWTRK
BYCONHO
BNPONS
BNPONEP
BNPONDA
BNPONBB
BNNWTRK
BNCONHO

107. DOGFITE

CALLS:
GETPTRS
UNSTAT
TAPE6#
OUTCI.
HISTORY
FINOBLK
DROPBK
DELADD
XPAA
RANQOM.
GINME
DESTROY

CALLED BY:
SELECT

108. DOGTHNK

CALLS:
MESBILD
DELADD
TAPE6#
OUTCI.
GOTOAB
FUELCHK

CALLED BY:
TFLYCRC

109. DROPBLK

CALLS:
ENTRYP
RELEASE
EXITP

CALLED BY:
SAMWYPE
NUKBLND
NOWUCIT
KILFLIT
INT2CRC
HEXMOVE
FLTWYPE
DOGFITE
CRCTRAK
CRCLOSS
CRCKIL
CRCDIES

BTNASIN
BADMOVE
ATTACK
AB2CRC

110. DROPP05

CALLS:

YANK
RELEASE
MESBILD
DELADD
SEEKENG
GIMME
DLYACT

CALLED BY:

SEEKTAC
SAMPRCM
BNCONTC
CHKLAST
BOCTINK
BNPONFO
BNPONEP
BNCONLS
BNCONHO
BNCMDPR

111. DROPPS2

CALLS:

YANK
RELEASE
MESBILD
DELADD
SEEKENG
GIMME
DLYACT

CALLED BY:

SEEKENG

112. ENGAGE

CALLS:

GETPTRS
TRYSHOT
TAPE6#
OUTCI.
GIMME
DELADD
UNPACK
PACK
AMMOCHK
RELEASE
HISTORY

CALLED BY:

SELECT

113. ENTRY#

CALLS:

MESSAGE
RITEI
RECEP
TAPE6#
OUTCI.
ROUTER
ITRAP
SECOND

CALLED BY:

LTRMRG
UNPACK
UOLLOAD
UNSNAP
TXYZHXL
TXYZHX
THKZXY
THRPLN
THM2PS
TGTGONE
SNAP
SELECT
SCHTAB
RLWAVE
RLTGTYP

RLTGAK
RLRAID
RLFMAKT
RLCORD
RLABOB
REVISE
RELIST
RELEASE
PTREE
PLANOUT
PLAN
PELADD
PACK
OUTA
NOWUCIT
LTREE
LOADPL
LINEX
KOMPARE
JUGGLE
JTJ
JGESUIT
INITACO
IJZHX
HXOGTS
HOLD
MLTPNT
HISTORY
HEXMULT
HEXMLT
HEXINV
HEXCHZ
HEXADD
GIMME
GETPTRS
GETHEX
FSOUMP
FORMTGT
FINDFLT
FINDBLK
FETCH
DROPBLK
DELADD
CREATE
CORBOUN
CONTROL
CLOSCOR
CENTER
CANOTGT
ADDBLOK
ABVSCOR
MADEM

114. ENTSTAT

CALLS:
TAPE6#

CALLED BY:
HALT

	OUTCI.	FSDUMP
115. EOF		CALLED BY:
		DBGREAD
116. EXITP	CALLS:	CALLED BY:
	SECOND	LTRMRG
	MESSAGE	UNPACK
	RITEI	UOLLOAD
	RECER	UNSNAP
	TAPE6#	TXYZHXL
	OUTCI.	TXYZHX
	ITRAP	THXZXY
	ICHECK	THTRPLN
		THM2PS
		TSTGONE
		SNAP
		SELECT
		SCHTAB
		RLWAVE
		RLTGTYP
		RLTGTA
		RLRAID
		RLFMAKT
		RLCORD
		RLABOB
		REVISE
		RELIST
		RELEASE
		PTREE
		PLANOUT
		PLAN
		PELADD
		PACK
		OUTA
		NOWUCIT
		LTREE
		LOADPL
		LIVEX
		KOMPAR
		JUGGLE
		JTJ
		JGESUIT
		INITACQ
		IJ2HX
		HXDGTS
		HOLD
		HLTPNT
		HISTORY
		HEXMULT
		HEXMLT
		HEXINV
		HEXCHZ
		HEXADD
		GIHME

GETPTRS
GETHEX
FSDUMP
FORMTGT
FINDFLT
FINDBLK
FETCH
DROPBLK
DELADD
CREATE
CORBOUN
CONTROL
CLOSCOR
CENTER
CANDTGT
ADDBLCK
ABVSCOR
MADEM

117. FELDEL

CALLS:
UNPACK
UNSNAP
RELEASE
TAPE6#
OUTCI.

CALLED BY:
SELECT

118. FETCH

CALLS:
ENTRYP
INPBI.
EXITP

CALLED BY:
MADEM

119. FILERUP

CALLS:
RANDOM.
TAPE6#
OUTCI.
GIMME
YANK
STICK
DELAOD

CALLED BY:
BYPNRRL
BNPNBB

120. FINOBLK

CALLS:
ENTRYP
ROUTER
MESSAGE
RITEI
CLIST
EXITP

CALLED BY:
TOWER
THTRPLN
SAMWYPE
REDEBRF
NUKBLND
NOWUCIT
NEWMOVE
KILFLIT
INT2CRC
HEXMOVE
GOGETEM
GNDLOOK
FORMTGT
FLTWYPE

FINDIT
 FINDFLT
 DOGFITE
 CRCTRAK
 CRCLOSS
 CRCKIL
 CRCEVNT
 CRCDIES
 CANDTGT
 BTNASIN
 BAOMOVE
 INTASIN
 ATTACK
 A92CRC

121. FINDFLT	CALLS: ENTRYP HEXDIST FINDBLK KOMPARE GIMME ADDBLOK EXITP	CALLED BY: FORMTGT
122. FINDIT	CALLS: FINDBLK GIMME ADDBLOK	CALLED BY: INTFIND
123. FIRECHK	CALLS: THX2XY SIN. COS. SQRT. ATAN2. ASIN.	CALLED BY: TRYSHOT
124. FLITE	CALLS: INTRFLY UNPACK COMMAND HEXCHZ OPTPTH RELEASF SIN. DELAOD	CALLED BY: FLY
125. FLTGEOM	CALLS: PACK HEXCHZ THH2PS ATAN2.	CALLED BY: GOGETEM ACFRAG
126. FLTWPE	CALLS: UNPACK	CALLED BY: KILFLIT

GOTOAB
COMMANDRELEASF
FINDBLK
TRACE
MESSAGE
RITEI
CLIST
FSDUMP
OROPBLK
PACK

127. FLY

CALLS:

GETPTRS
UNSTAT
HEXMOVF
FUELCHK
SHRKILL
FLITE
DELADD
STATPAK
GIMME
PACK
TAPE6#
OUTCI.
DESTROY

CALLED BY:

SELECT

128. FLYSEF

CALLS:

DELADD
GIMME
ADDBLCK
TAPE6#
OUTCI.
MESBILD
RELEASE
HEXDIST
DETECT
HISTORY

CALLED BY:

CFLYCRC

129. FORMTGT

CALLS:

ENTRYP
GIMME
FINDBLK
FINDFLT
RELEASE
EXITP

CALLED BY:

CANDTGT

130. FSDUMP

CALLS:

ENTRYP
TRACE
ENTSTAT
CLIST
RITER
LNPLT
LCMLC
TAPE6#
OUTCI.

CALLED BY:

YANK
UNSTAT
TRKCHK
JTJ
FLTYPE

	OUTCR. EXITP	
131. FSINIT		CALLED BY: MADEM
132. FUELCHK	CALLS: UNPACK HXDIST GOTOAB TAPE6# OUTCI. MESBILD DELAOD	CALLED BY: FLY DOGTHNK
133. GETHEX	CALLS: ENTRYP HXDGTS MESSAGE RITEI GIMME SCHTAB EXITP	CALLED BY: TTY2HXL SCHEDUL PTRAND NUKBLND NOWUCIT HEXCHZ
134. GETPTRS	CALLS: ENTRYP EXITP	CALLED BY: TOWER RONDSEE PONDER PLAN PERCEPT SAMPRCH FLY ENGAGE DOGFITE DESTROY BYPASUP BYENOPS ATTACK ASSIGN
135. GIMME	CALLS: ENTRYP HALT MESSAGE RITEI RITEP EXITP	CALLED BY: WTHDRAW UOLLOAD UMPIRE TOWER TJADIL T3TGONE SHRKILL SEEKTAC SEEKENG SOIGEST SAMWYPE REVISE RESUPPLY RELOAD

RELEASE
REDEBRF
BYALCOV
PELADD
OPTPTH
NUKBLNO
NOWUCIT
NEWPERC
NEWMOVE
BYCONTC
BNCONTC
NAYBOR
MESBILD
KILFLIT
INT2CRC
INITACQ
GOGETEM
GETHEX
FORMTGT
FLYSEE
FLY
FINDIT
FINDFLT
FILERUP
ENGAGE
DROPPS2
DROPP0S
DOGFITE
DESTROY
DELADD
CRFLTML
CREATE
CRCSEE
CRCEVNT
COVAPLY
CORBOUN
CHKCOV
CANOTGT
BYUPDAT
BYPONTM
BYPASUP
BYNWTRK
BYCONMD
BNPONS
BNPONEP
BNPONDA
BNPONBB
BNNWTRK
BNCONMD
BATTOUT
BATCEAS
BADMOVE
ATKASES
ALLOPAT
ALLOFU

ALLUBAT
ACFRAG
ACCEPT
ABVSCOR
AB2CRC

136. GNDLOOK	CALLS: FINOBLK XPD RANDOM. DELADD HISTORY TAPE6# OUTCI. UNPACK THM2PS ATAN2.	CALLED BY: CFLYCRC
137. GOGETEM	CALLS: UNPACK FINOBLK CRFLTML TAPE6# OUTCI. GIMME AOOBLOK PTRAND PACK FLTGEOM UOLLOAD INITACO DELADD MESBILD	CALLED BY: TOWER
138. GOTOAB	CALLS: UNPACK FLTWYPE OPTPTH RELEASE THM2PS ATAN2.	CALLED BY: FUELCHK DOOTHNK
139. HALT	CALLS: HOLD OUTCI. RITER RITEI TRACE RECER ENTSTAT CLIST PAGE ADUMP ISOUMP ENOFIL.	CALLED BY: UNPACK ITRAP RECCON SELECT RELEASE PACK HLTPTNT GIMME MADEM

STOP.

140. HANDZPT	CALLS: BATCEAS YANK STICK TOADIL RELOCAT DILOUT	CALLED BY: BNLALLE
141. HEXA00	CALLS: ENTRYP IT0J. EXITP	CALLED BY: SCHEDUL PTRAND NUKBLND NOWUCIT HEXMLT HEXCHZ
142. HEXCHZ	CALLS: ENTRYP TRACE MESSAGE RITEP CLIST HEXA00 HEXINV GETHEX EXITP	CALLED BY: OPTPTH FLTGEOM FLITE
143. HEX0IST		CALLLED BY: SCHEDUL NAYBOR INTRFLY FUELCHK FLYSEE FINDFLY DETECT CRCEVNT CORBOUN CLOSCOR BTNASIN BADMOVE INTASIN AIRTHNK ACFRAG
144. HEXINV	CALLS: ENTRYP IT0J. EXITP	CALLED BY: HEXCHZ
145. HEXMLT	CALLS: HEXA00 ENTRYP HX0GTS	CALLED BY: IJ2HX

	MESSAGE RITEI EXITP	
146. HEXMOVE	CALLS: TRACE MESSAGE RITEI CLIST UNPACK XTOI. NOWUCIT FINOBLK DROPBK RELEASE PACK UOLLOAD	CALLED BY: FLY
147. HEXMULT	CALLS: ENTRYP ITOJ. EXITP	CALLED BY: PTRAND NOWUCIT
148. HISTORY	CALLS: ENTRYP MESSAGE MASKER TAPE6# OUTCI. EXITP	CALLED BY: UMPIRE SHRKILL NEWPERC NEWMOVE GNDLOOK FLYSEE ENGAGE DOGFITE CRFLTML BADMOVE ATTACK
149. HLTPNT	CALLS: ENTRYP SECOND HALT EXITP	CALLED BY: CONTROL
150. HOLD	CALLS: ENTRYP OUTBI. REWIND. EXITP	CALLED BY: HALT
151. HXDGTS	CALLS: ENTRYP EXITP	CALLED BY: TH2HX THX2XY HEXMLT GETHEX
152. HXMLT2	CALLS:	

	TH2HX IJ2HX	
153. ICHECK	CALLS: OUTCI.	CALLED BY: EXITP
154. IJ2HX	CALLS: HEXMLT ENTRYP ITOU. EXITP	CALLED BY: XY2HX TXY2HXL HXMLT2
155. INIT		CALLED BY: MADEM
156. INITACQ	CALLS: ENTRYP GIMME ADOBLOK ALOG. MESSAGE RITEI XTOI. NOWUCIT EXITP	CALLED BY: TOWER GOGETEM
157. INRANGE	CALLS: THM2PS ATAN2. SQRT. COS. TAPE6# OUTCI. AZILIM	CALLED BY: BYCONTC CHKCOV BYPONRL BYNWTRK BNPONBB
158. INSECT	CALLS: TAN. ATAN2.	CALLED BY: AZILIM
159. INSERT	CALLS: UNPACK XSHIFT	CALLED BY: SSLL
160. INTASIN	CALLS: FINOBLK TRACE MESSAGE RITEI CLIST UNPACK HEXOIST TGTHX MESBILD PACK DELAOD	CALLED BY: ASSIGN

		TAPE6# OUTCI. BTNASIN	
161. INTFIND	CALLS:	UNPACK TAPE6# OUTCI. FINDIT PACK MESBILD DELOAD	
162. INTRFLY	CALLS:	TGTMEY HEXDIST MESBILD DELOAD TAPE6# OUTCI. UNPACK THH2PS ATAN2. OPTPTH RELEASE	CALLED BY: FLITE
163. INT2CRC	CALLS:	UNPACK FINDBLK RELEASE CRCKIL TAPE6# OUTCI. DETECT MESBILD DELOAD DROPBLK GIMME ADDBLOK	CALLED BY: CRCTHNK
164. IPJL	CALLS:	LCMLOC ISHIFT TAPE6# OUTCI.	
165. ISGUMP	CALLS:	OUTCI. OUTCR.	CALLED BY: HALT UNPACK PACK
166. ISHIFT			CALLED BY: SLL PTREE LOSRADR

		JTJ IPJL
167. ITRAP	CALLS: HALT	CALLED BY: EXITP ENTRYP THTRPLN
168. JGESUIT	CALLS: ENTRYP THX2XY ATAN2. EXITP	CALLED BY: CANDTGT
169. JTJ	CALLS: ENTRYP MESSAGE RITEI ISHIFT TAPE6# OUTCI. FSDUMP STOP. EXITP	
170. JUGGLE	CALLS: ENTRYP EXITP	CALLED BY: THX2XY
171. KILFLIT	CALLS: GIMME DELAOD STOP. UNPACK RELEASE UNSTAT FLTWYPE FINDBLK DROPSLK	CALLED BY: DESTROY
172. KOMPARE	CALLS: ENTRYP UNPACK PACK EXITP	CALLED BY: THTRPLN FINOFLT
173. LCMLOC		CALLED BY: UNPACK RITEP PACK IPJL FSDUMP MADEN
174. LINEX	CALLS:	CALLED BY:

	ENTRYP EXITP	OPTPTH CORBOUN
175. LNPLOT	CALLS: TAPE6# OUTCI.	CALLED BY: CLIST2 OUTPTRS FSDUMP CLIST
176. LOADPL	CALLS: ENTRYP EXITP	
177. LOSRAOR	CALLS: ISHIFT OPTPTH THM2PS SQRT. COS. UNPACK RELEASE	CALLED BY: BYTKCHK DETECT CRCEVNT
178. LRKPRS		CALLED BY: MADEM
179. LTREE	CALLS: ENTRYP RELEASE LTRMRG EXITP	CALLED BY: CONTROL
180. LTRMRG	CALLS: ENTRYP EXITP	CALLED BY: LTREE DELADD
181. MADEM	CALLS: QINTRY. RECCON ENTRYP INPFI. DBGREAD RECOVR GOTOER. FETCH PAGE LCMLOC FSINIT OTHRDAT DISPDAT HALT INIT LRKPRS RELIST DELADD CONTROL	

EXITP
END.

182. MASKER

CALLED BY:
HISTORY

183. MESSAGE

CALLS:
TAPE6#
OUTCI.

CALLLED BY:
CLIST2
EXITP
ENTRYP
UNSTAT
TGTHEX
SELECT
RELEASE
PACK
OUTPTRS
TRKCHK
SAMPRCM
NAYBOR
JTJ
INITAC3
HISTORY
HEXMOVE
HEXMLT
HEXCHZ
GIMME
GETHEX
FLTWYPE
FINDBLK
DELADD
CRCDIES
CLIST
INTASIN

184. MESBILD

CALLS:
GIMME
PACK

CALLLED BY:
TOWER
BYCONTC
BNLALLE
INTRFLY
INT2CRC
INTFIND
CRC2INT
GOGETEM
FUELCHK
FLYSEE
DROPPS2
DROPPS
DGGTHNK
DECRALD
CRCTRAK
COMMAND
BYPONRL
BYPONFD
BYPONER
SYNWTRK

BYHEDUP
 BYCONLS
 BTRYTNK
 BTNASIN
 BOCTINK
 BNPONFD
 BNPONEP
 BNCONLS
 BNCONHD
 BNCMOPR
 BADMOVE
 INTASIN
 AMMOCHK
 ALLOBAT
 AIRTHNK

185. NAYBOR	CALLS: UNPACK MESSAGE RITEI TRACE CLIST HEXOIST GIMME DELADD RELEASE	CALLED BY: SELECT
186. NEWMOVE	CALLS: FINDBLK RANDOM. TAPE6M OUTCI. GIMME ADDBLK HISTORY DELADD	CALLED BY: CRCTRAK
187. NEWPERC	CALLS: HISTORY GIMME STICK DELADD	CALLED BY: SAMPRCH BYPASUP
188. NOWUCIT	CALLS: ENTRYP HEXADD HEXMUL GETHE^ GIMME ADDBLK FINDBLK DROPBLK RELEASE EXITP	CALLED BY: TERMAC3 INITAC3 HEXMOVE

189. NUKBLND	CALLS: UNPACK GIMME HEXADD GETHEX TAPE6# OUTCI. DELADD SAMWYPE TERMACO FINDBLK RELEASE DROPBLK UNLINK	CALLED BY: UMPIRE ATTACK
190. OPTPTH	CALLS: GIMME THM2PS HEXCHZ LINEX PACK	CALLED BY: RONOSEE LOSRAOR INTRFLY CRCZINT GOTOAB FLITE CORBOUN ACFRAG
191. OTHROAT		CALLED BY: MAOEM
192. OUTA	CALLS: ENTRYP TAPE6# OUTCI. EXITP	CALLED BY: YANK CHKCOV ALLOBAT ACCEPT
193. OUTPTRS	CALLS: LNPLOT MESSAGE RITEI RITER RITP UNPACK	
194. PACK	CALLS: ENTRYP LCMLOC PAGE MESSAGE RITEI TRACE ROUTER RITP ISDUMP HALT EXITP	CALLED BY: YANK WITHDRAW UMPIRE TOWER STICK SHRKILL SETASSN SEEKTAC SDIGEST PTPNER BYALCOV

PREPAFU
 OPTPTH
 MESBILD
 KOMPARE
 INTFIND
 HEXMOVE
 GOGETEM
 FLY
 FLTWYPE
 FLTGEOM
 ENGAGE
 OLYACT
 CRFLTML
 COMMO
 BYUPDAT
 BYPONTM
 BYPONER
 BYPASUP
 BTVASIN
 BNRECOV
 BNPONER
 BNCONHO
 INTASIN
 ALLOBAT
 ACFRAG
 ACCEPT

195. PAGE

CALLS:
 TAPE6#
 OUTCI.

CALLED BY:
 CLIST2
 HALT
 PACK
 CLIST
 MAOEM

196. PATDEC

CALLS:
 CANCEL0
 SEEXTFU

CALLED BY:
 BYALCOV

197. PELADD

CALLS:
 ENTRYP
 GIMME
 PTRMRG
 EXITP

CALLED BY:
 REDEBRF
 CANDTGT
 AVAILBL

198. PERCEPT

CALLS:
 GETPTRS
 ABSEE
 CFLYCRC
 SAMSEE

CALLED BY:
 SELECT

199. PLAN

CALLS:
 ENTRYP
 GETPTRS
 THTRPLN
 EXITP

CALLED BY:
 SELECT

200. PLANOUT	CALLS: ENTRYP OUTCI. EXITP	CALLED BY: THTRPLV
201. PONDER	CALLS: GETPTRS TFLYCRC BOCTINK BTRYTNK	CALLED BY: SELECT
202. PREPAFU	CALLS: UNPACK PACK AUTOPRI STICK RELOCAT DELAOD	CALLED BY: BYCONTC BYPONRL BYNWTRK
203. PRIORITY		CALLED BY: SETASSN CHKLAST BNPONBB BNCONHO ALLOBAT
204. PTPONER	CALLS: UNPACK PACK CANCALO SEEKTFU OILOUT DELAOD	CALLED BY: BYPONER
205. PTRAND	CALLS: UNPACK RANDOM. HEXADD HEXMULT GETHEX	CALLED BY: GOGETEM
206. PTREE	CALLS: ENTRYP RELEASE ISHIFT EXITP	CALLED BY: TGTGONE CANOTGT
207. PTRMRG		CALLED BY: PELAOD
208. READIL	CALLS: UNPACK YANK STICK	CALLED BY: SEEKTAC CANCALO BNPONEP

	TAPE6# OUTCI. RELOCAT DELAOD	
209. RECCON	CALLS: TAPE6# OUTCI. ADUMP HALT	CALLED BY: MADEM
210. RECER	CALLS: TAPE6# OUTCI. OUTCR.	CALLED BY: HALT EXITP ENTRYP DELAOD
211. RECOVR		CALLED BY: MADEM
212. REDEBRF	CALLS: UNPACK FINDBLK WIPEOUT GIMME ADDBLOK PELAOD DESTROY	CALLED BY: TOWER
213. RELEASE	CALLS: ENTRYP MESSAGE RITEI RITEP HALT GIMME EXITP	CALLED BY: WITHDRAW UMPIRE TERMACQ SEEKTAC SEEKENG SOIGEST SAMWYPE SAMSEE SAMATON RONDSEE RLWAVE RLTGTYP RLTGTA RLRAID RLFMAKT RLCORD RLABDB REVISE RELSILL RELIST PTREE BYALCOV NUKBLNO SAMPRCH NOUCIT

NAYBOR
L TREE
LOSRAOR
KILFLIT
INTRFLY
INT2CRC
CRC2INT
HEXMOVE
GOTOAB
FORMTGT
FLYSEE
FLTWYPE
FLITE
FEDEL
ENGAGE
DROPPS2
DROPPS
DROPBLK
DILOUT
DESTROY
CRCTHNK
CRCSEE
CRCKIL
CRCEVNT
CRCOIES
COVAPLY
CONTROL
COMMO
CNACTTK
CANCALO
BYUPOAT
BYPONTH
BYPONRS
BYPONFD
BYPONER
BYHEDUP
BYENOPS
BYCMOPR
BTN2CRC
BTNASIN
BOCTINK
BNPONFD
BNPONDA
BNPONBD
BNNOTRO
BNCONHD
BATTOUT
BATCEAS
BAOMOVE
AVAILBL
ACFRAG
ACCEPT
A3SEE
A32CRC

214. RELIST	CALLS: ENTRYP RELEASE EXITP	CALLED BY: MADEM
215. RELOAD	CALLS: GIMME GOTOER. DELAOD RESUPLY	CALLED BY: BYPONRS
216. RELOCAT	CALLS: YANK STICK	CALLED BY: TOADIL SETASSN SAMATON READIL PREPAFU HANDZPT BYCONTC CHKLAST BYNWRK BYHEDUP BYCONHO BNPONBB
217. RELSILL	CALLS: RELEASE	
218. RENDEVU	CALLS: UNPACK THXZY TXYZHXL	CALLED BY: SCHEDUL
219. RESUPLY	CALLS: GIMME DELAOD	CALLED BY: RELOAD
220. REVISE	CALLS: ENTRYP GIMME ABVSCOR RELEASE EXITP	CALLED BY: THTRPLY
221. RITEI	CALLS: TAPE6# OUTCI.	CALLED BY: CLIST2 HALT EXITP ENTRYP TSTHEX SELECT RELEASE PACK OUTPTRS

		TRKCHK NAYBOR JTJ INITACQ HEXMOVE HEXMLT GIMME GETHEX FLTWYPE FINDBLK CRCOIES CLIST INTASIN
222. RITEP	CALLS: LCMLDC TAPE6# OUTCI.	CALLED BY: CLIST2 UNPACK RELEASE PACK OUTPTRS SAMPRCM HEXCHZ GIMME CLIST
223. RITER	CALLS: TAPE6# OUTCI.	CALLED BY: CLIST2 HALT OUTPTRS FSDUMP CLIST
224. RLABDB	CALLS: ENTRYP RELEASE EXITP	CALLED BY: RLCORD
225. RLCORD	CALLS: ENTRYP RLABDB RELEASE EXITP	CALLED BY: RLRAID
226. RLFMAKT	CALLS: ENTRYP RELEASE EXITP	CALLED BY: RLTGAK
227. RLRAID	CALLS:	CALLED BY: BYPONER BATCEAS
245. SELECT	CALLS:	CALLED BY:

CONTROL

ENTRYP
 GOTOER.
 MESSAGE
 RITEI
 HALT
 FELOEL
 ASSIGN
 ATTACK
 COMMO
 DOGFITE
 ENGAGE
 FLY
 NAYBOR
 PERCEPT
 PLAN
 PONDER
 TOWER
 UMPIRE
 TAPE6#
 OUTCI.
 OUTCR.
 EXITP

246. SETASSN

CALLS:

UNPACK
 PACK
 STICK
 PRIORITY
 RELOCAT
 DELA00

CALLED BY:

BNCONTC
 BNPONDA
 BNPONBB
 BNNWTRK

247. SHRKILL

CALLS:

ALOG.
 RANDOM.
 HISTORY
 UNPACK
 TAPE6#
 OUTCI.
 GIMME
 PACK
 DELA00
 DESTROY

CALLED BY:

FLY
 ATTACK

248. SHUFFLE

CALLED BY:

TH2HX

249. SKSBTRK

CALLS:

UNPACK
 SEEKP

CALLED BY:

BNPONEP

250. SNAP

CALLS:

ENTRYP
 EXITP

CALLED BY:

DELA00
 CONTROL

251. SSSL

CALLS:

CALLED BY:

	UNPACK IShift INSERT	WTHORAW SEEKTAC SEEKENG BYALCOV BYUPDAT BATTOUT
252. STATPAK		CALLED BY: TFLYCRC RONDSEE FLY CFLYCRC ATTACK
253. STICK	CALLS: UNPACK PACK	CALLED BY: TOADIL SETASSN SOIGEST READIL PREPAFU TRKCHK RELOCAT HANDZPT NEWPERC BYTKCHK BYCONTC BNLALLE BNCONTC FILERUP OLYACT BYNWTRK BYCONHO BNPONEP BNPONBB BNCONHO BATCEAS ALLOPAT ALLOFU ALLOBAT ACCEPT
254. TERMADO	CALLS: UNPACK XTOI. NOWUCIT RELEASE	CALLED BY: NUKBLND DESTROY
255. TFLYCRC	CALLS: CRCTHMK UNSTAT AIRTHNK DOGTHNK STATPAK	CALLED BY: PONDER
256. TGTGONE	CALLS:	CALLED BY:

	ENTRYP PTREE GIMME EXITP	CANDTGT
257. TGTHX	CALLS: TRACE MESSAGE RITEI CLIST UNPACK TMM2PS ATAN2. COS. SQRT. SIN. TXY2HXL	CALLED BY: INTRFLY BADMOVE INTASIN AIRTHNK
258. TMM2PS	CALLS: ENTRYP THX2XY EXITP	CALLED BY: UMPIRE TGTHX RONDSEE OPTPTH TRKCHK INRANGE LOSRADR INTRFLY CRC2INT GOTOAB GNDLOOK FLTGEOM DETECT CORBOUN COMMAND ATKASES
259. THTRPLN	CALLS: ENTRYP CORBOUN REVISE FINOBLK KOMPARE CANDTGT ITRAP AVAILBL SCHEDUL DELADD PLANOUT RLRAID EXITP	CALLED BY: PLAN
260. THX2XY	CALLS: ENTRYP HXDGTS JUGGLE	CALLED BY: TMM2PS RENDEUV JGESUIT

	EXITP	FIRECHK
261. TH2HX	CALLS: HXDGTS SHUFFLE	CALLED BY: HXMLT2
262. TOADIL	CALLS: TAPE6# OUTCI. YANK STICK RELOCAT GIMME	CALLED BY: SDIGEST SAMATON TRKCHK HANDZPT BYTKCHK BYCONTC BNCONTC DILOUT ALLOBAT
263. TOWER	CALLS: GETPTRS TAPE6# OUTCI. UNPACK UOLLOAD INITACO DELADD PACK FINDBLK REDEBRF GIMME GOGETEM MESBILD DESTROY	CALLED BY: SELECT
264. TRACE	CALLS: TRPRRT TRPRMT TRPRNT	CALLED BY: HALT YANK UNSTAT TGTHX PACK SAMPRCM NAYBOR HEXMOVE HEXCHZ FSDUMP FLTWYPE DELADD CRCOIES INTASIN
265. TRKCHK	CALLS: THH2PS SQRT. ATAN2. YANK MESSAGE	CALLED BY: SEEKTFU BATTCOV

RITEI
CLIST
FSDUMP
STICK
TOADIL
BNLALLE

266. TRPRMT		CALLED BY: TRACE
267. TRPRNT	CALLS: TAPE6# OUTCI.	CALLED BY: TRACE
268. TRPRRT		CALLED BY: TRACE
269. TRYSHOT	CALLS: GOTOER. TAPE6# OUTCI. YANK BATCEAS BNLALLE FIRECHK	CALLED BY: ENGAGE BYPONTM
270. TXYZHX	CALLS: ENTRYP EXITP	
271. TXYZHL	CALLS: ENTRYP COS. SIN. XTOI. CENTER IJZHX GETHEX EXITP	CALLED BY: TGTHEX RENDEVU CORBOUN
272. UMPIRE	CALLS: UNPACK THM2PS SORT. XTOI. HISTORY RANDOM. TAPE6# OUTCI. GIMME PACK DELAOD DESTROY RELEASE NUKBLND	CALLED BY: SELECT

273. UNLINK

CALLS:
UNPACKCALLED BY:
NUKBLND
DESTROY

274. UNPACK

CALLS:
ENTRYP
LCMLOC
RITEP
ISDUMP
HALT
EXITPCALLED BY:
YANK
WTHORAW
UNSTAT
UNLINK
UMPIRE
TOWER
TGTHEX
TERMACQ
STICK
SSLL
SKSBTRK
SHRKILL
SETASSN
SEEKTAC
SDIGEST
SCHEDUL
SAMWYPE
SAMATON
RONOSEE
RENDEVU
REDEBRF
READIL
PTRAND
PTPONER
BYALCOV
PREPAFU
OUTPTRS
NUKBLND
BYCONTC
NAYBOR
LOSRADR
KOMPARC
KILFLIT
INTRFLY
INT2CRC
INTFINO
CRC2INT
INSERT
HEXMOVE
GOTOAB
BOGETEM
GNLOOK
FUELCHK
FLTWYPE
FLITE
FEDEL
ENGAGE
DLYACT
OILOUT

DETECT
 CRFLTML
 CRCTRAX
 CRCTHNK
 CRCSEE
 CRCLOSS
 CRCKIL
 CRCEVNT
 CRCDIES
 COMMO
 COMMAND
 CHKCOV
 BYUPDAT
 BYPONTM
 BYPONRS
 BYPONRL
 BYPONER
 BYPASUP
 BYNOTRO
 BYHEDUP
 BYENDPS
 BTN2CRC
 BTNASIN
 BNRECOV
 BNPNSS
 BVPONEP
 BVPONDA
 BVPONBB
 BATTOUT
 BATCEAS
 BAOMOVE
 INTASIN
 ACFRAG
 ACCEPT
 ABSEE
 AB2CRC

275. UNSNAP

CALLS:
 ENTRYP
 EXITP

CALLED BY:
 FELDEL
 CONTROL

276. UNSTAT

CALLS:
 TRACE
 MESSAGE
 CLIST
 FSDUMP
 STOP.
 UNPACK

CALLED BY:
 TFLYCRC
 RONDSEE
 KILFLIT
 FLY
 DOGFITE
 DESTROY
 CFLYCRC
 ATTACK

277. UOLLOAD

CALLS:
 ENTRYP
 GIMME
 ADDBLOK

CALLED BY:
 TOWER
 HEXMOVE
 GOGETEM

EXITP

278. WIPEOUT

CALLED BY:
REDEBRF

279. #THORAW

CALLS:
UNPACK
GIMME
PACK
SLL
DECRALO
SEEKENG
DLYACT
RELEASE

CALLED BY:
BNRECOV

280. XPAA

CALLED BY:
DGGFITE

281. XPO

CALLED BY:
GNOLOOK

282. XPK

CALLED BY:
ATTACK

283. XSHIFT

CALLED BY:
INSERT

284. XY2HX

CALLS:
CENTER
IJ2HX

285. YANK

CALLS:
UNPACK
TAPE6#
OUTCI.
TRACE
OUTA
FSDUMP
STOP.
PACK

CALLED BY:
TRYSHOT
TOADIL
SEEKTAC
SDIGEST
READIL
TRKCHK
SAMPRCM
RELOCAT
HANDZPT
BYTKCHK
BYCONTC
BNLALLE
BYCONTC
FILERUP
DROPPS2
DROPPS
DILOUT
CVACTTK
CANCALO
BYRONFD
BYENOPS
BTRYTNK
BNRONFD

PAGE 63

BNPONEP
BNPONDA
BNCONHO
BATTOUT
ALLOBAT

FORTRAN LIBRARY REFERENCE LIST - MAJEM

PAGE 54

1. ALOG.	<p> CALLED BY: SHRKILL INITACQ </p>
2. ASIN.	<p> CALLED BY: FIRECHK </p>
3. ATAN2.	<p> CALLED BY: TGTHX RNDSEE TRKCHK INRANGE BYTKCHK JGESUIT INTRFLY CRCZINT INSECT GOTOAB GNDLOOK FLTGEOM FIRECHK DETECT COMMAND ATKASES </p>
4. COS.	<p> CALLED BY: TXYZHL TGTHX INRANGE LOSRADP FIRECHK DETECT CRCEVNT CORBOUN AZILIM </p>
5. DECODI.	<p> CALLED BY: OBGREAD </p>
6. ENDFIL.	<p> CALLED BY: MALT </p>
7. END.	<p> CALLED BY: MAJEM </p>
8. GOTOER.	<p> CALLED BY: TRYSHOT SELECT SAMWYPE RELOAD COMMAND </p>

BYPNRL
BYPNSS
BYPONEP
BYPONDA
BYPONBB
BYCMOPR
AMMOCHK
MADEM

9. INPBI.

CALLED BY:
FETCH

10. INPCI.

CALLED BY:
DBGREAD

11. INPFI.

CALLED BY:
MADEM

12. ITOJ.

CALLED BY:
IJ2HX
HEXMULT
HEXINV
HEXADD
DGTSHX

13. OUTBI.

CALLED BY:
HOLD

14. OUTCI.

CALLED BY:
CLIST2
ISDUMP
MALT
ADUMP
DBGREAD
ICHECK
DISPPYB
DISPPRO
DISPPAY
DISPPAF
DISPFMF
DISPFLT
DISPFDB
DISPAQD
DISPADS
DISPACR
DISPACL
DISPACO
DISPABQ
DISPDAT
EXITP
ENTRYP
ENTSTAT
ROUTER
RECER
RECCON

YANK
UMPIRE
TRYSHOT
TRPRNT
TOWER
TOADIL
SHRKILL
SELECT
SDIGEST
SAMATON
RONDSEE
RITER
RITEP
RITEI
READIL
BYALCOV
PLANOUT
PAGE
OUTA
NUKBLNO
INRANGE
NEWMOVE
BNCONTC
MESSAGE
LNPLOT
JTJ
IPJL
INTRFLY
INTZCRC
INTFIND
CRC2INT
HISTORY
GOGETEM
GNDLOOK
FUELCHK
FSDUMP
FLYSEE
FLY
FILERUP
FELDEL
ENGAGE
DOGTHNK
DOGFITE
DETECT
DESTROY
DECRALO
CRCTRAK
CRCKIL
CRCEVNT
COMMAND
CLIST
CHKCOV
CANCALO
BYPONTM
BYPONRS

BYPNRL
BYNOTRO
BYCONHO
BYCMOPR
BTRYTNK
BTN2CRC
BTNASIN
BOCTINK
BNPONFO
BNPONFA
BNPONBD
BNPONBS
BNNOTRO
BNCONHO
INTASIN
ATKASES
AMMOCHK
ALLOPAT
ALLOFU
ALLOBAT
AIRTHNK
ACCEPT
ABSEE
AB2CRC

15. OUTCR.

CALLED BY:

CLIST2
ISDUMP
ADUMP
ROUTER
RECER
SELECT
FSOUMP
CLIST

16. QINTRY.

CALLED BY:

MADEM

17. RANDOM.

CALLED BY:

UMPIRE
SHRKILL
SDIGEST
SCHEDUL
PTRAND
NEWMOVE
GNOLOOK
FILERUP
DOGFITE
DETECT
CRCTRK
CRCEVNT
CANDTGT
ATTACK
AIRTHNK

18. REWIND.

CALLED BY:
HOLD

19. SIN.

CALLED BY:
TXYZHXL
TSTHGX
FLITE
FIRECHK
CORBOUN
AZILIM

20. SORT.

CALLED BY:
UMPIRE
TSTHGX
TRKCHK
INRANGE
BYTKCHK
LOSRADE
FIRECHK
AZILIM

21. STOP.

CALLED BY:
HALT
YANK
UNSTAT
KILFI IT
JTJ
ACCEPT

22. TAN.

CALLED BY:
INSECT

23. TAPE6*

CALLED BY:
CLIST2
EXITP
ENTRYP
ENTSTAT
ROUTER
RECER
RECCON
YANK
UMPIRE
TRYSHOT
TRPRNT
TOWER
TOADIL
SHRKILL
SELECT
SOIGEST
SAMATON
RONDSEE
RITER
RITEP
RITEI

READIL
BYALCOV
PAGE
OUTA
NUKBLND
INRANGE
NEWMOVE
BNCONTC
MESSAGE
LNPLOT
JTJ
IPJL
INTRFLY
INT2CRC
INTFIND
CRC2INT
HISTORY
GOGETEM
GNOLOOK
FUELCHK
FSOUMP
FLYSEE
FLY
FILERUP
FELDEL
ENGAGE
DOGTHNK
DOGFITE
DETECT
DESTROY
DECRALO
CRCTRAX
CRCKIL
CRCEVNT
COMMAND
CLIST
CHKCOV
CANCALO
BYPONTM
BYPONRS
BYPONRL
BYNOTRO
BYCONHO
BYCMOPR
BTRYTNK
BTN2CRC
BTNASIN
BOCTINK
BNPONFD
BNPONFA
BNPONBD
BNPONBB
BNNOTRO
BNCONHO
INTASIN

ATKASES
AMMOCHK
ALLOPAT
ALLOFU
ALLOBAT
AIRTHNK
ACCEPT
ABSEE
ABZCRC

24. XTOI.

CALLED BY:

UMPIRE
TXYZHL
TERMACQ
INITACQ
HEXMOVE

```

00001      1. MADEM
00002          2. QINTRY.
00003          2. RECCON
00004              3. TAPE6#
00005              3. OUTCI.
00006              3. ADUMP
00007                  4. OUTCI.
00008                  4. OUTCR.
00009          3. HALT
00010              4. HOLD
00011                  5. ENTRYP
00012                      6. MESSAGE
00013                          7. TAPES#
00014                          7. OUTCI.
00015                      6. RITEI
00016                          7. TAPES#
00017                          7. OUTCI.
00018                      6. RECER
00019                          7. TAPES#
00020                          7. OUTCI.
00021                          7. OUTCR.
00022                      6. TAPE6#
00023                      6. OUTCI.
00024                      6. ROUTER
00025                          7. TAPES#
00026                          7. OUTCI.
00027                          7. OUTCR.
00028                      6. ITRAP
00029                          7. HALT
00030                              (SEE LINE 00009)
00031                      6. SECOND
00032          5. OUTBI.
00033          5. REWIND.
00034          5. EXITP
00035              6. SECOND
00036              6. MESSAGE
00037              6. RITEI
00038              6. RECER
00039              6. TAPE6#
00040              6. OUTCI.
00041              6. ITRAP
00042              6. ICHECK
00043                  7. OUTCI.
00044          4. RITER
00045              5. TAPE6#
00046              5. OUTCI.
00047          4. RITEI
00048              5. TRACE
00049                  5. TRPRRT
00050                  5. TRPRMT
00051                  5. TRPRNT
00052              6. TAPE6#
    
```

```

00053          5. OUTCI.
00054          4. RECER          (SEE LINE 00018)
00055          4. ENTSTAT
00056          5. TAPE6#
00057          5. OUTCI.
00058          4. CLIST
00059          5. PAGE
00060          6. TAPE6#
00061          6. OUTCI.
00062          5. LNPLOT
00063          6. TAPE6#
00064          6. OUTCI.
00065          5. MESSAGE      (SEE LINE 00012)
00066          5. RITEP
00067          6. LCMLOC
00068          6. TAPE6#
00069          6. OUTCI.
00070          5. RITER        (SEE LINE 00044)
00071          5. RITEI        (SEE LINE 00015)
00072          5. TAPE6#
00073          5. OUTCI.
00074          5. OUTCR.
00075          5. CLIST2
00076          6. PAGE          (SEE LINE 00059)
00077          6. LNPLOT      (SEE LINE 00062)
00078          6. MESSAGE      (SEE LINE 00012)
00079          6. RITEI        (SEE LINE 00015)
00080          6. RITER        (SEE LINE 00044)
00081          6. TAPE6#
00082          6. OUTCI.
00083          6. OUTCR.
00084          6. RITEP        (SEE LINE 00066)
00085          4. PAGE          (SEE LINE 00059)
00086          4. ADUMP         (SEE LINE 00006)
00087          4. ISDUMP
00088          5. OUTCI.
00089          5. OUTCR.
00090          4. ENOFIL.
00091          4. STOP.
00092          2. ENTRYP      (SEE LINE 00011)
00093          2. INPFI.
00094          2. DB3READ
00095          3. OUTCI.
00096          3. INPCI.
00097          3. EOF
00098          3. DECOOI.
00099          2. RECOVR
00100          2. GOTOER.
00101          2. FETCH
00102          3. ENTRYP      (SEE LINE 00011)
00103          3. INPFI.
00104          3. EXITP        (SEE LINE 00033)
00105          2. PAGE          (SEE LINE 00059)
00106          2. LCMLOC
00107          2. FSINIT

```

```

00108      2. JTHROAT
00109      2. DISPOAT
00110      3. OUTCI.
00111      3. DISPADS
00112      4. OUTCI.
00113      3. DISPF08
00114      4. OUTCI.
00115      4. DISPFMF
00116      5. OUTCI.
00117      5. DISPFLT
00118      6. OUTCI.
00119      6. DISPPAY
00120      7. OUTCI.
00121      7. DISPPYB
00122      8. OUTCI.
00123      6. DISPAQD
00124      7. OUTCI.
00125      6. DISPACO
00126      7. OUTCI.
00127      6. DISPPRO
00128      7. OUTCI.
00129      3. DISPFLT (SEE LINE 00117)
00130      3. DISPACO (SEE LINE 00125)
00131      3. DISPPAF
00132      4. OUTCI.
00133      4. DISPPYB (SEE LINE 00121)
00134      3. DISPPRO (SEE LINE 00127)
00135      3. DISPAQD (SEE LINE 00123)
00136      3. DISPA3Q
00137      4. OUTCI.
00138      3. DISPACR
00139      4. OUTCI.
00140      4. DISPAQL
00141      5. OUTCI.
00142      2. HALT (SEE LINE 00009)
00143      2. INIT
00144      2. LRKPRS
00145      2. RELIST
00146      3. ENTRYP (SEE LINE 00011)
00147      3. RELEASE
00148      4. ENTRYP (SEE LINE 00011)
00149      4. MESSAGE (SEE LINE 00012)
00150      4. RITEI (SEE LINE 00015)
00151      4. RITEP (SEE LINE 00066)
00152      4. HALT (SEE LINE 00009)
00153      4. GIMME
00154      5. ENTRYP (SEE LINE 00011)
00155      5. HALT (SEE LINE 00009)
00156      5. MESSAGE (SEE LINE 00012)
00157      5. RITEI (SEE LINE 00015)
00158      5. RITEP (SEE LINE 00066)
00159      5. EXITP (SEE LINE 00033)
00160      4. EXITP (SEE LINE 00033)
00161      3. EXITP (SEE LINE 00033)
00162      2. DELADD

```

00163	3. ENTRYP	(SEE LINE 00011)
00164	3. MESSAGE	(SEE LINE 00012)
00165	3. TRACE	(SEE LINE 00048)
00166	3. RECFR	(SEE LINE 00018)
00167	3. GINME	(SEE LINE 00153)
00168	3. SNAP	
00169	4. ENTRYP	(SEE LINE 00011)
00170	4. EXITP	(SEE LINE 00033)
00171	3. LTRMRG	
00172	4. ENTRYP	(SEE LINE 00011)
00173	4. EXITP	(SEE LINE 00033)
00174	3. EXITP	(SEE LINE 00033)
00175	2. CONTROL	
00176	3. ENTRYP	(SEE LINE 00011)
00177	3. LTRMRG	
00178	4. ENTRYP	(SEE LINE 00011)
00179	4. RELEASE	(SEE LINE 00147)
00180	4. LTRMRG	(SEE LINE 00171)
00181	4. EXITP	(SEE LINE 00033)
00182	3. SELECT	
00183	4. ENTRYP	(SEE LINE 00011)
00184	4. GOTDER.	
00185	4. MESSAGE	(SEE LINE 00012)
00186	4. RITEI	(SEE LINE 00015)
00187	4. HALT	(SEE LINE 00009)
00188	4. FELDEL	
00189	5. UNPACK	
00190	6. ENTRYP	(SEE LINE 00011)
00191	6. LCMLOC	
00192	6. RITEP	(SEE LINE 00066)
00193	6. ISDUMP	(SEE LINE 00087)
00194	6. HALT	(SEE LINE 00009)
00195	6. EXITP	(SEE LINE 00033)
00196	5. UNSNAP	
00197	6. ENTRYP	(SEE LINE 00011)
00198	6. EXITP	(SEE LINE 00033)
00199	5. RELEASE	(SEE LINE 00147)
00200	5. TAPE6#	
00201	5. OUTCI.	
00202	ASSIGN	
00203	5. GETPTRS	
00204	6. ENTRYP	(SEE LINE 00011)
00205	6. EXITP	(SEE LINE 00033)
00206	5. INTASIN	
00207	6. FINDBLK	
00208	7. ENTRYP	(SEE LINE 00011)
00209	7. ROUTER	(SEE LINE 00024)
00210	7. MESSAGE	(SEE LINE 00012)
00211	7. RITEI	(SEE LINE 00015)
00212	7. CLIST	(SEE LINE 00058)
00213	7. EXITP	(SEE LINE 00033)
00214	6. TRACE	(SEE LINE 00048)
00215	6. MESSAGE	(SEE LINE 00012)
00216	6. RITEI	(SEE LINE 00015)
00217	6. CLIST	(SEE LINE 00058)

00218	6. UNPACK	(SEE LINE 00189)
00219	6. HEXDIST	
00220	6. TGTTEX	
00221	7. TRACE	(SEE LINE 00048)
00222	7. MESSAGE	(SEE LINE 00012)
00223	7. RITEI	(SEE LINE 00015)
00224	7. CLIST	(SEE LINE 00058)
00225	7. UNPACK	(SEE LINE 00189)
00226	7. THM2PS	
00227	8. ENTRYP	(SEE LINE 00011)
00228	8. THX2XY	
00229	9. ENTRYP	(SEE LINE 00011)
00230	9. HXOGTS	
00231	10. ENTRYP	(SEE LINE 00011)
00232	10. EXITP	(SEE LINE 00033)
00233	9. JGGLE	
00234	10. ENTRYP	(SEE LINE 00011)
00235	10. EXITP	(SEE LINE 00033)
00236	9. EXITP	(SEE LINE 00033)
00237	8. EXITP	(SEE LINE 00033)
00238	7. ATAN2.	
00239	7. COS.	
00240	7. SQRT.	
00241	7. SIN.	
00242	7. TXY2MXL	
00243	8. ENTRYP	(SEE LINE 00011)
00244	8. COS.	
00245	8. SIN.	
00246	8. XTQI.	
00247	8. CENTER	
00248	9. ENTRYP	(SEE LINE 00011)
00249	9. EXITP	(SEE LINE 00033)
00250	8. IJ2MX	
00251	9. HEXMLT	
00252	10. HEXADD	
00253	11. ENTRYP	(SEE LINE 00011)
00254	11. ITOJ.	
00255	11. EXITP	(SEE LINE 00033)
00256	10. ENTRYP	(SEE LINE 00011)
00257	10. HXOGTS	(SEE LINE 00230)
00258	10. MESSAGE	(SEE LINE 00012)
00259	10. RITEI	(SEE LINE 00015)
00260	10. EXITP	(SEE LINE 00033)
00261	9. ENTRYP	(SEE LINE 00011)
00262	9. ITOJ.	
00263	9. EXITP	(SEE LINE 00033)
00264	8. GETTEX	
00265	9. ENTRYP	(SEE LINE 00011)
00266	9. HXOGTS	(SEE LINE 00230)
00267	9. MESSAGE	(SEE LINE 00012)
00268	9. RITEI	(SEE LINE 00015)
00269	9. GIMME	(SEE LINE 00153)
00270	9. SCHATB	
00271	10. ENTRYP	(SEE LINE 00011)
00272	10. EXITP	(SEE LINE 00033)

```

00273          9. EXITP          (SEE LINE 00033)
00274          8. EXITP          (SEE LINE 00033)
00275      6. MESBILD
00276          7. GIMME          (SEE LINE 00153)
00277          7. PACK
00278          8. ENTRYP          (SEE LINE 00011)
00279          8. LCMLJC
00280          8. PAGE           (SEE LINE 00059)
00281          8. MESSAGE        (SEE LINE 00012)
00282          8. RITEI          (SEE LINE 00015)
00283          8. TRACEI        (SEE LINE 00048)
00284          8. ROUTER        (SEE LINE 00024)
00285          8. RITEP          (SEE LINE 00066)
00286          8. ISDUMP        (SEE LINE 00087)
00287          8. HALT          (SEE LINE 00009)
00288          8. EXITP          (SEE LINE 00033)
00289      6. PACK              (SEE LINE 00277)
00290      6. DELADD           (SEE LINE 00162)
00291      6. TAPESW
00292      6. OUTCI.
00293      6. BTNASIN
00294          7. UNPACK          (SEE LINE 00189)
00295          7. FINDBLK        (SEE LINE 00207)
00296          7. PACK           (SEE LINE 00277)
00297          7. HEXDIST
00298          7. RELEASE        (SEE LINE 00147)
00299          7. DROPBLK
00300          8. ENTRYP          (SEE LINE 00011)
00301          8. RELEASE        (SEE LINE 00147)
00302          8. EXITP          (SEE LINE 00033)
00303          7. MESBILD        (SEE LINE 00275)
00304          7. DELADD         (SEE LINE 00162)
00305          7. TAPESW
00306          7. OUTCI.
00307
00308      4. ATTACK
00309      5. GETPTRS          (SEE LINE 00203)
00310      5. UNSTAT
00311          6. TRACE          (SEE LINE 00048)
00312          6. MESSAGE        (SEE LINE 00012)
00313          6. CLIST          (SEE LINE 00058)
00314          6. FSDUMP
00315          7. ENTRYP          (SEE LINE 00011)
00316          7. TRACE          (SEE LINE 00048)
00317          7. ENTSTAT        (SEE LINE 00055)
00318          7. CLIST          (SEE LINE 00058)
00319          7. RITER          (SEE LINE 00044)
00320          7. LNPLDT        (SEE LINE 00062)
00321          7. LCMLJC
00322          7. TAPESW
00323          7. OUTCI.
00324          7. OUTCR.
00325          7. EXITP          (SEE LINE 00033)
00326      6. STOP.
00327      6. UNPACK          (SEE LINE 00189)
5. SHRKILL

```

00328	6. ALOG.	
00329	6. RANDOM.	
00330	6. HISTORY	
00331	7. ENTRYP	(SEE LINE 00011)
00332	7. MESSAGE	(SEE LINE 00012)
00333	7. MASKER	
00334	7. TAPES#	
00335	7. OUTCI.	
00336	7. EXITP	(SEE LINE 00033)
00337	6. UNPACK	(SEE LINE 00189)
00338	6. TAPE6#	
00339	6. OUTCI.	
00340	6. GIMME	(SEE LINE 00153)
00341	6. PACK	(SEE LINE 00277)
00342	6. DELA00	(SEE LINE 00162)
00343	6. DESTROY	
00344	7. TAPES#	
00345	7. OUTCI.	
00346	7. GETPTRS	(SEE LINE 00203)
00347	7. TERMACQ	
00348	8. UNPACK	(SEE LINE 00189)
00349	8. XT0I.	
00350	8. NO#UCIT	
00351	9. ENTRYP	(SEE LINE 00011)
00352	9. HEXADD	(SEE LINE 00252)
00353	9. HEXMULT	
00354	10. ENTRYP	(SEE LINE 00011)
00355	10. ITOJ.	
00356	10. EXITP	(SEE LINE 00033)
00357	9. GETHEX	(SEE LINE 00264)
00358	9. GIMME	(SEE LINE 00153)
00359	9. ADDBLOK	
00360	10. ENTRYP	(SEE LINE 00011)
00361	10. EXITP	(SEE LINE 00033)
00362	9. FINDBLK	(SEE LINE 00207)
00363	9. DROPBLK	(SEE LINE 00299)
00364	9. RELEASE	(SEE LINE 00147)
00365	9. EXITP	(SEE LINE 00033)
00366	8. RELEASE	(SEE LINE 00147)
00367	7. KILFLIT	
00368	8. GIMME	(SEE LINE 00153)
00369	8. DELA00	(SEE LINE 00162)
00370	8. STOP.	
00371	8. UNPACK	(SEE LINE 00189)
00372	8. RELEASE	(SEE LINE 00147)
00373	8. UNSTAT	(SEE LINE 00309)
00374	8. FLTTYPE	
00375	9. UNPACK	(SEE LINE 00189)
00376	9. RELEASE	(SEE LINE 00147)
00377	9. FINDBLK	(SEE LINE 00207)
00378	9. TRACE	(SEE LINE 00048)
00379	9. MESSAGE	(SEE LINE 00012)
00380	9. RITEI	(SEE LINE 00015)
00381	9. CLIST	(SEE LINE 00058)
00382	9. FSDJMP	(SEE LINE 00313)

00383	9. DROPBLK	(SEE LINE 00299)
00384	9. PACK	(SEE LINE 00277)
00385	8. FINDBLK	(SEE LINE 00207)
00386	8. DROPBLK	(SEE LINE 00299)
00387	7. GIMME	(SEE LINE 00153)
00388	7. DELADD	(SEE LINE 00162)
00389	7. SAMWYPE	
00390	8. UNPACK	(SEE LINE 00189)
00391	8. GIMME	(SEE LINE 00153)
00392	8. DELADD	(SEE LINE 00162)
00393	8. QOTDER.	
00394	8. RELEASE	(SEE LINE 00147)
00395	8. BYENDPS	
00396	9. SEEKP	
00397	9. UNPACK	(SEE LINE 00189)
00398	9. YANK	
00399	10. JNPACK	(SEE LINE 00189)
00400	10. TAPE6#	
00401	10. OUTCI.	
00402	10. TRACE	(SEE LINE 00048)
00403	10. OUTA	
00404	11. ENTRYP	(SEE LINE 00011)
00405	11. TAPE6#	
00406	11. OUTCI.	
00407	11. EXITP	(SEE LINE 00033)
00408	10. FSDUMP	(SEE LINE 00313)
00409	10. STOP.	
00410	10. PACK	(SEE LINE 00277)
00411	9. RELEASE	(SEE LINE 00147)
00412	9. GETPTRS	(SEE LINE 00203)
00413	9. CRCLOSS	
00414	10. FINDBLK	(SEE LINE 00207)
00415	10. JNPACK	(SEE LINE 00189)
00416	10. DROPBLK	(SEE LINE 00299)
00417	10. DELADD	(SEE LINE 00162)
00418	9. BVLALLE	
00419	10. MESSILJ	(SEE LINE 00275)
00420	10. DELADD	(SEE LINE 00162)
00421	10. YANK	(SEE LINE 00398)
00422	10. HANDZPT	
00423	11. BATCEAS	
00424	12. CANCELQ	
00425	13. YANK	(SEE LINE 00398)
00426	13. RELEASE	(SEE LINE 00147)
00427	13. TAPE6#	
00428	13. OUTCI.	
00429	13. READIL	
00430	14. UNPACK	(SEE LINE 0018)
00431	14. YANK	(SEE LINE 0039)
00432	14. STICK	
00433	15. UNPACK	(SEE LINE 0)
00434	15. PACK	(SEE LINE 0)
00435	14. TAPE6#	
00436	14. OUTCI.	
00437	14. RELOCAT	

PAGE 79

00438		15. YANK	(SEE LINE 0
00439		15. STICK	(SEE LINE 0
00440		14. DELA00	(SEE LINE 0016
00441		12. GIMME	(SEE LINE 00153)
00442		12. UNPACK	(SEE LINE 00189)
00443		12. SEEKTFU	
00444		13. TRKCHK	
00445		14. THH2PS	(SEE LINE 0022
00446		14. SQRT.	
00447		14. ATAN2.	
00448		14. YANK	(SEE LINE 0039
00449		14. MESSAGE	(SEE LINE 0001
00450		14. RITEI	(SEE LINE 0001
00451		14. CLIST	(SEE LINE 0005
00452		14. FSDUMP	(SEE LINE 0031
00453		14. STICK	(SEE LINE 0043
00454		14. TOADIL	
00455		15. TAPE6#	
00456		15. OUTCI.	
00457		15. YANK	(SEE LINE 0
00458		15. STICK	(SEE LINE 0
00459		15. RELOCAT	(SEE LINE 0
00460		15. GIMME	(SEE LINE 0
00461		14. BNLALLE	(SEE LINE 0041
00462		13. ALLOFU	
00463		14. GIMME	(SEE LINE 0015
00464		14. STICK	(SEE LINE 0043
00465		14. DELA00	(SEE LINE 0016
00466		14. TAPE6#	
00467		14. OUTCI.	
00468		13. ALLOPAT	
00469		14. GIMME	(SEE LINE 0015
00470		14. STICK	(SEE LINE 0043
00471		14. DELA00	(SEE LINE 0016
00472		14. TAPE6#	
00473		14. OUTCI.	
00474		12. STICK	(SEE LINE 00432)
00475		12. RELEASE	(SEE LINE 00147)
00476		11. YANK	(SEE LINE 00399)
00477		11. STICK	(SEE LINE 00432)
00478		11. TOADIL	(SEE LINE 00454)
00479		11. RELOCAT	(SEE LINE 00437)
00480		11. DILOUT	
00481		12. CNACTK	
00482		13. YANK	(SEE LINE 00398)
00483		13. RELEASE	(SEE LINE 00147)
00484		12. RELEASE	(SEE LINE 00147)
00485		12. UNPACK	(SEE LINE 00189)
00486		12. YANK	(SEE LINE 00399)
00487		12. TOADIL	(SEE LINE 00454)
00488		10. DILOUT	(SEE LINE 00480)
00489		10. STICK	(SEE LINE 00432)
00490		8. FIV0BLK	(SEE LINE 00207)
00491		8. DRO3BLK	(SEE LINE 00299)
00492		7. CRC0IES	

00493	8. UNPACK	(SEE LINE 00189)
00494	8. RELEASE	(SEE LINE 00147)
00495	8. DELADD	(SEE LINE 00162)
00496	8. FINDBLK	(SEE LINE 00207)
00497	8. TRACE	(SEE LINE 00048)
00498	8. MESSAGE	(SEE LINE 00012)
00499	8. RITEI	(SEE LINE 00015)
00500	8. CLIST	(SEE LINE 00058)
00501	8. DROPBLK	(SEE LINE 00299)
00502	7. UNLINK	
00503	8. UNPACK	(SEE LINE 00189)
00504	7. RELEASE	(SEE LINE 00147)
00505	7. UNSTAT	(SEE LINE 00309)
00506	5. FINDBLK	(SEE LINE 00207)
00507	5. XPK	
00508	5. RANDOM.	
00509	5. NUKBLND	
00510	6. UNPACK	(SEE LINE 00189)
00511	6. GIMME	(SEE LINE 00153)
00512	6. HEXADD	(SEE LINE 00252)
00513	6. GETHEX	(SEE LINE 00264)
00514	6. TAPE6#	
00515	6. OUTCI.	
00516	6. DELADD	(SEE LINE 00162)
00517	6. SAMWPE	(SEE LINE 00389)
00518	6. TERMACB	(SEE LINE 00347)
00519	6. FINDBLK	(SEE LINE 00207)
00520	6. RELEASE	(SEE LINE 00147)
00521	6. DROPBLK	(SEE LINE 00299)
00522	6. UNLINK	(SEE LINE 00502)
00523	5. DROPBLK	(SEE LINE 00299)
00524	5. HISTORY	(SEE LINE 00330)
00525	5. DELADD	(SEE LINE 00162)
00526	5. STATPAK	
00527	4. COMMO	
00528	5. UNPACK	(SEE LINE 00189)
00529	5. DELADD	(SEE LINE 00162)
00530	5. RELEASE	(SEE LINE 00147)
00531	5. PACK	(SEE LINE 00277)
00532	4. 003FITE	
00533	5. GETPTRS	(SEE LINE 00203)
00534	5. UNSTAT	(SEE LINE 00309)
00535	5. TAPE6#	
00536	5. OUTCI.	
00537	5. HISTORY	(SEE LINE 00330)
00538	5. FINDBLK	(SEE LINE 00207)
00539	5. DROPBLK	(SEE LINE 00299)
00540	5. DELADD	(SEE LINE 00162)
00541	5. XPAA	
00542	5. RANDOM.	
00543	5. GIMME	(SEE LINE 00153)
00544	5. DESTROY	(SEE LINE 00343)
00545	4. ENGAGE	
00546	5. GETPTRS	(SEE LINE 00203)
00547	5. TRYSHOT	

00548	6. GOTOER.	
00549	6. TAPE6#	
00550	6. OUTCI.	
00551	6. YANK	(SEE LINE 00398)
00552	6. BATCEAS	(SEE LINE 00423)
00553	6. BNLALLE	(SEE LINE 00418)
00554	6. FIRECHK	
00555	7. THX2XY	(SEE LINE 00228)
00556	7. SIN.	
00557	7. COS.	
00558	7. SQRT.	
00559	7. ATAN2.	
00560	7. ASIN.	
00561	5. TAPE6#	
00562	5. OUTCI.	
00563	5. GIMME	(SEE LINE 00153)
00564	5. DELADD	(SEE LINE 00162)
00565	5. UNPACK	(SEE LINE 00189)
00566	5. PACK	(SEE LINE 00277)
00567	5. AMMOCHK	
00568	6. MESBILD	(SEE LINE 00275)
00569	6. DELADD	(SEE LINE 00162)
00570	6. GOTOER.	
00571	6. BYNOTRD	
00572	7. UNPACK	(SEE LINE 00189)
00573	7. BATCEAS	(SEE LINE 00423)
00574	7. OILOJT	(SEE LINE 00480)
00575	7. TAPES#	
00576	7. OUTCI.	
00577	6. TAPES#	
00578	6. OUTCI.	
00579	5. RELEASE	(SEE LINE 00147)
00580	5. HISTORY	(SEE LINE 00330)
00581	4. FLY	
00582	5. GETPTRS	(SEE LINE 00203)
00583	5. UNSTAT	(SEE LINE 00309)
00584	5. HEXMOVE	
00585	6. TRACE	(SEE LINE 00048)
00586	6. MESSAGE	(SEE LINE 00012)
00587	6. RITEI	(SEE LINE 00015)
00588	6. CLIST	(SEE LINE 00058)
00589	6. UNPACK	(SEE LINE 00189)
00590	6. XTOI.	
00591	6. NOWUCIT	(SEE LINE 00350)
00592	6. FINDBLK	(SEE LINE 00207)
00593	6. DROPBLK	(SEE LINE 00299)
00594	6. RELEASE	(SEE LINE 00147)
00595	6. PACK	(SEE LINE 00277)
00596	6. UOLLOAD	
00597	7. ENTRYP	(SEE LINE 00011)
00598	7. GIMME	(SEE LINE 00153)
00599	7. ADDBLOK	(SEE LINE 00359)
00600	7. EXITP	(SEE LINE 00033)
00601	5. FUELCHK	
00602	6. UNPACK	(SEE LINE 00189)

00603	6. HEXDIST	
00604	6. GOTOAB	
00605	7. UNPACK	(SEE LINE 00189)
00606	7. FLTWYPE	(SEE LINE 00374)
00607	7. OPTPTH	
00608	8. GIMMEI	(SEE LINE 00153)
00609	8. THM2PS	(SEE LINE 00226)
00610	8. HEXCHZ	
00611	9. ENTRYP	(SEE LINE 00011)
00612	9. TRACE	(SEE LINE 00048)
00613	9. MESSAGE	(SEE LINE 00012)
00614	9. RITEP	(SEE LINE 00066)
00615	9. CLIST	(SEE LINE 00058)
00616	9. HEXADD	(SEE LINE 00252)
00617	9. HEXINV	
00618	10. ENTRYP	(SEE LINE 00011)
00619	10. ITOJ.	
00620	10. EXITP	(SEE LINE 00033)
00621	9. GETHEX	(SEE LINE 00264)
00622	9. EXITP	(SEE LINE 00033)
00623	8. LIVEX	
00624	9. ENTRYP	(SEE LINE 00011)
00625	9. EXITP	(SEE LINE 00033)
00626	8. PACK	(SEE LINE 00277)
00627	7. RELEASE	(SEE LINE 00147)
00628	7. THM2PS	(SEE LINE 00226)
00629	7. ATAN2.	
00630	6. TAPE6N	
00631	6. OUTCI.	
00632	6. MESBILD	(SEE LINE 00275)
00633	6. DELADD	(SEE LINE 00162)
00634	5. SHRKILL	(SEE LINE 00327)
00635	5. FLITE	
00636	6. INTRFLY	
00637	7. TGTHEX	(SEE LINE 00220)
00638	7. HEXDIST	
00639	7. MESBILD	(SEE LINE 00275)
00640	7. DELADD	(SEE LINE 00162)
00641	7. TAPE6N	
00642	7. OUTCI.	
00643	7. UNPACK	(SEE LINE 00189)
00644	7. THM2PS	(SEE LINE 00226)
00645	7. ATAN2.	
00646	7. OPTPTH	(SEE LINE 00607)
00647	7. RELEASE	(SEE LINE 00147)
00648	6. UNPACK	(SEE LINE 00189)
00649	6. COMMAND	
00650	7. UNPACK	(SEE LINE 00189)
00651	7. GOTOER.	
00652	7. DELADD	(SEE LINE 00162)
00653	7. TAPE6N	
00654	7. OUTCI.	
00655	7. MESBILD	(SEE LINE 00275)
00656	7. FLTWYPE	(SEE LINE 00374)
00657	7. THM2PS	(SEE LINE 00226)

00658	7. ATAN2.	
00659	6. HEXCHZ	(SEE LINE 00610)
00660	6. OPTPTH	(SEE LINE 00607)
00661	6. RELEASE	(SEE LINE 00147)
00662	6. SIN.	
00663	6. DELADD	(SEE LINE 00162)
00664	5. DELADD	(SEE LINE 00162)
00665	5. STATPAK	
00666	5. GIMME	(SEE LINE 00153)
00667	5. PACK	(SEE LINE 00277)
00668	5. TAPE6#	
00669	5. OUTCI.	
00670	5. DESTROY	(SEE LINE 00343)
00671	4. NAYBOR	
00672	5. UNPACK	(SEE LINE 00189)
00673	5. MESSAGE	(SEE LINE 00012)
00674	5. RITEI	(SEE LINE 00015)
00675	5. TRACE	(SEE LINE 00048)
00676	5. CLIST	(SEE LINE 00058)
00677	5. HEXDIST	
00678	5. GIMME	(SEE LINE 00153)
00679	5. DELADD	(SEE LINE 00162)
00680	5. RELEASE	(SEE LINE 00147)
00681	4. PERCEPT	
00682	5. GETPTRS	(SEE LINE 00203)
00683	5. ABSEE	
00684	6. UNPACK	(SEE LINE 00189)
00685	6. DELADD	(SEE LINE 00162)
00686	6. RELEASE	(SEE LINE 00147)
00687	6. TAPE6#	
00688	6. OUTCI.	
00689	5. CFLYCRC	
00690	6. CRCSEE	
00691	7. RELEASE	(SEE LINE 00147)
00692	7. GIMME	(SEE LINE 00153)
00693	7. UNPACK	(SEE LINE 00189)
00694	7. DELADD	(SEE LINE 00162)
00695	7. CRCEVNT	
00696	8. GIMME	(SEE LINE 00153)
00697	8. TAPE6#	
00698	8. OUTCI.	
00699	8. DELADD	(SEE LINE 00162)
00700	8. RELEASE	(SEE LINE 00147)
00701	8. FIND3LK	(SEE LINE 00207)
00702	8. DETECT	
00703	9. HEXDIST	
00704	9. UNPACK	(SEE LINE 00189)
00705	9. THM2PS	(SEE LINE 00226)
00706	9. ATAN2.	
00707	9. LOGRADR	
00708	10. ISHIFT	
00709	10. OPTPTH	(SEE LINE 00607)
00710	10. THM2PS	(SEE LINE 00226)
00711	10. SORT.	
00712	10. COS.	

00713	
00714	
00715	
00716	
00717	
00718	
00719	
00720	
00721	
00722	
00723	
00724	
00725	
00726	
00727	
00728	
00729	
00730	
00731	
00732	
00733	
00734	
00735	
00736	
00737	
00738	
00739	
00740	
00741	
00742	
00743	
00744	
00745	
00746	
00747	
00748	
00749	
00750	
00751	
00752	
00753	
00754	
00755	
00756	
00757	
00758	
00759	
00760	
00761	
00762	
00763	
00764	
00765	
00766	
00767	
	10. JNPACK (SEE LINE 00189)
	10. RELEASE (SEE LINE 00147)
	9. CJS.
	9. RANDOM.
	9. TAPE#
	9. OUTCI.
	8. UNPACK (SEE LINE 00189)
	8. HEXDIST
	8. LJSRADR (SEE LINE 00707)
	8. CJS.
	8. RANDOM.
	8. CRCTRK
	9. FINDBLK (SEE LINE 00207)
	9. UNPACK (SEE LINE 00189)
	9. MESSILD (SEE LINE 00275)
	9. DELADD (SEE LINE 00162)
	9. TAPE#
	9. OUTCI.
	9. CRCKIL
	10. FINDBLK (SEE LINE 00207)
	10. JNPACK (SEE LINE 00189)
	10. RELEASE (SEE LINE 00147)
	10. DROPBLK (SEE LINE 00299)
	10. TAPE#
	10. OUTCI.
	10. DELADD (SEE LINE 00162)
	9. BADMOVE
	10. FINDBLK (SEE LINE 00207)
	10. GIMME (SEE LINE 00153)
	10. ADDBLOK (SEE LINE 00359)
	10. HISTORY (SEE LINE 00330)
	10. UNPACK (SEE LINE 00189)
	10. RELEASE (SEE LINE 00147)
	10. DROPBLK (SEE LINE 00299)
	10. TGTTEX (SEE LINE 00220)
	10. HEXDIST
	10. MESSILD (SEE LINE 00275)
	10. DELADD (SEE LINE 00162)
	9. DROPBLK (SEE LINE 00299)
	9. RANDOM.
	9. NEWMOVE
	10. FINDBLK (SEE LINE 00207)
	10. RANDOM.
	10. TAPES#
	10. OUTCI.
	10. GIMME (SEE LINE 00153)
	10. ADDBLOK (SEE LINE 00359)
	10. HISTORY (SEE LINE 00330)
	10. DELADD (SEE LINE 00162)
	(SEE LINE 00309)
4. UNSTAT	
5. FLYSEE	
	7. DELADD (SEE LINE 00162)
	7. GIMME (SEE LINE 00153)
	7. ADDBLOK (SEE LINE 00359)
	7. TAPES#

00768	7. OUTCI.	
00769	7. MESBILD	(SEE LINE 00275)
00770	7. RELEASE	(SEE LINE 00147)
00771	7. MEXOIST	
00772	7. DETECT	(SEE LINE 00702)
00773	7. HISTORY	(SEE LINE 00330)
00774	6. ATKASES	
00775	7. GIMME	(SEE LINE 00153)
00776	7. DELAJO	(SEE LINE 00162)
00777	7. DESTROY	(SEE LINE 00343)
00778	7. TAPES#	
00779	7. OUTCI.	
00780	7. ADDBLK	(SEE LINE 00359)
00781	7. THM2PS	(SEE LINE 00226)
00782	7. ATAN2.	
00783	6. CRC2INT	
00784	7. UNPACK	(SEE LINE 00189)
00785	7. TAPES#	
00786	7. OUTCI.	
00787	7. MESBILD	(SEE LINE 00275)
00788	7. DELAJO	(SEE LINE 00162)
00789	7. OPTPT4	(SEE LINE 00607)
00790	7. RELEASE	(SEE LINE 00147)
00791	7. THM2PS	(SEE LINE 00226)
00792	7. ATAN2.	
00793	6. RONDSEE	
00794	7. UNPACK	(SEE LINE 00189)
00795	7. OPTPT4	(SEE LINE 00607)
00796	7. THM2PS	(SEE LINE 00226)
00797	7. ATAN2.	
00798	7. RELEASE	(SEE LINE 00147)
00799	7. TAPES#	
00800	7. OUTCI.	
00801	7. GETPTRS	(SEE LINE 00203)
00802	7. UNSTAT	(SEE LINE 00309)
00803	7. STATPAK	
00804	6. GNOLOOK	
00805	7. FINDBLK	(SEE LINE 00207)
00806	7. XPD	
00807	7. RANDOM.	
00808	7. DELAJO	(SEE LINE 00162)
00809	7. HISTORY	(SEE LINE 00330)
00810	7. TAPES#	
00811	7. OUTCI.	
00812	7. UNPACK	(SEE LINE 00189)
00813	7. THM2PS	(SEE LINE 00226)
00814	7. ATAN2.	
00815	6. STATPAK	
00816	5. SAMSEE	
00817	6. SAMPRCM	
00818	7. TRACE	(SEE LINE 00048)
00819	7. MESSAGE	(SEE LINE 00012)
00820	7. RITEP	(SEE LINE 00066)
00821	7. CLIST	(SEE LINE 00058)
00822	7. SEEK?	

00823	7. DETECT	(SEE LINE 00702)
00824	7. BYTKCHK	
00825	8. SPT.	
00826	8. ATAN2.	
00827	8. LJSR4OR	(SEE LINE 00707)
00828	8. DELADD	(SEE LINE 00162)
00829	8. YANK	(SEE LINE 00398)
00830	8. STICK	(SEE LINE 00432)
00831	8. TADIL	(SEE LINE 00454)
00832	7. NEWPERC	
00833	8. HISTORY	(SEE LINE 00330)
00834	8. GIMME	(SEE LINE 00153)
00835	8. STICK	(SEE LINE 00432)
00836	8. DELADD	(SEE LINE 00162)
00837	7. GETPTRS	(SEE LINE 00203)
00838	7. CRCTRK	(SEE LINE 00724)
00839	7. BYPASS	
00840	8. SEEKPI	
00841	8. UNPACK	(SEE LINE 00189)
00842	8. GIMME	(SEE LINE 00153)
00843	8. PACK	(SEE LINE 00277)
00844	8. NEWPERC	(SEE LINE 00832)
00845	8. GETPTRS	(SEE LINE 00203)
00846	8. CRCTRK	(SEE LINE 00724)
00847	7. BYENOPS	(SEE LINE 00395)
00848	7. CRCLJSS	(SEE LINE 00413)
00849	7. DROPPS	
00850	8. YANK	(SEE LINE 00398)
00851	8. RELEASE	(SEE LINE 00147)
00852	8. MESBILD	(SEE LINE 00275)
00853	8. DELADD	(SEE LINE 00162)
00854	8. SEEKENG	
00855	9. DROPPS2	
00856	10. YANK	(SEE LINE 00398)
00857	10. RELEASE	(SEE LINE 00147)
00858	10. MESBILD	(SEE LINE 00275)
00859	10. DELADD	(SEE LINE 00162)
00860	10. SEEKENG	(SEE LINE 00854)
00861	10. GIMME	(SEE LINE 00153)
00862	10. DLYACT	
00863	11. UNPACK	(SEE LINE 00189)
00864	11. PACK	(SEE LINE 00277)
00865	11. STICK	(SEE LINE 00432)
00866	9. GIMME	(SEE LINE 00153)
00867	9. DELADD	(SEE LINE 00162)
00868	9. SLL	
00869	10. UNPACK	(SEE LINE 00189)
00870	10. XSHIFT	
00871	10. INSERT	
00872	11. UNPACK	(SEE LINE 00189)
00873	11. XSHIFT	
00874	9. ALLOBAT	
00875	10. PRIORITY	
00876	10. MESBILD	(SEE LINE 00275)
00877	10. GIMME	(SEE LINE 00153)

00878		10. PACK	(SEE LINE 00277)
00879		10. TAPE6#	
00880		10. OUTCI.	
00881		10. JUTA	(SEE LINE 00403)
00882		10. DELADD	(SEE LINE 00162)
00883		10. YANK	(SEE LINE 00398)
00884		10. STICK	(SEE LINE 00432)
00885		10. TOADIL	(SEE LINE 00454)
00886		9. RELEASE	(SEE LINE 00147)
00887		8. GIMME	(SEE LINE 00153)
00888		8. DLYACT	(SEE LINE 00862)
00889		7. BNLALLE	(SEE LINE 00418)
00890		7. YANK	(SEE LINE 00398)
00891		7. RELEASE	(SEE LINE 00147)
00892		6. DELADD	(SEE LINE 00162)
00893		6. RELEASE	(SEE LINE 00147)
00894	4. PLAN		
00895	5. ENTRYP	(SEE LINE 00011)	
00896	5. GETPTRS	(SEE LINE 00203)	
00897	5. THTRPLN		
00898	6. ENTRYP	(SEE LINE 00011)	
00899	6. CORBOUN		
00900	7. ENTRYP	(SEE LINE 00011)	
00901	7. GIMME	(SEE LINE 00153)	
00902	7. THW2PS	(SEE LINE 00226)	
00903	7. TXY24XL	(SEE LINE 00242)	
00904	7. HEXDIST		
00905	7. SIN.		
00906	7. COS.		
00907	7. LINEX	(SEE LINE 00623)	
00908	7. OPTPTH	(SEE LINE 00607)	
00909	7. EXITP	(SEE LINE 00033)	
00910	6. REVISE		
00911	7. ENTRYP	(SEE LINE 00011)	
00912	7. GIMME	(SEE LINE 00153)	
00913	7. ABVSCOR		
00914	8. ENTRYP	(SEE LINE 00011)	
00915	8. CLJSCOR		
00916	9. ENTRYP	(SEE LINE 00011)	
00917	9. HEXDIST		
00918	9. EXITP	(SEE LINE 00033)	
00919	8. GIMME	(SEE LINE 00153)	
00920	8. AJOBLK	(SEE LINE 00359)	
00921	8. EXITP	(SEE LINE 00033)	
00922	7. RELEASE	(SEE LINE 00147)	
00923	7. EXITP	(SEE LINE 00033)	
00924	6. FINDBLK	(SEE LINE 00207)	
00925	6. KOMPARE		
00926	7. ENTRYP	(SEE LINE 00011)	
00927	7. UNPACK	(SEE LINE 00189)	
00928	7. PACK	(SEE LINE 00277)	
00929	7. EXITP	(SEE LINE 00033)	
00930	6. CANDTGT		
00931	7. ENTRYP	(SEE LINE 00011)	
00932	7. FINDBLK	(SEE LINE 00207)	

00933	7. CLOSCOR	(SEE LINE 00915)
00934	7. JGESJIT	
00935	8. ENTRYP	(SEE LINE 00011)
00936	8. THXZY	(SEE LINE 00228)
00937	8. ATAN2.	
00938	8. EXITP	(SEE LINE 00033)
00939	7. FORMTGT	
00940	8. ENTRYP	(SEE LINE 00011)
00941	8. GIMME	(SEE LINE 00153)
00942	8. FINDBLK	(SEE LINE 00207)
00943	8. FINDFLT	
00944	9. ENTRYP	(SEE LINE 00011)
00945	9. HEXDIST	
00946	9. FINDBLK	(SEE LINE 00207)
00947	9. KMPARE	(SEE LINE 00925)
00948	9. GIMME	(SEE LINE 00153)
00949	9. AJOBLOK	(SEE LINE 00359)
00950	9. EXITP	(SEE LINE 00033)
00951	8. RELEASE	(SEE LINE 00147)
00952	8. EXITP	(SEE LINE 00033)
00953	7. GIMME	(SEE LINE 00153)
00954	7. ADDBLOK	(SEE LINE 00359)
00955	7. TGTGJNE	
00956	8. ENTRYP	(SEE LINE 00011)
00957	8. PTREE	
00958	9. ENTRYP	(SEE LINE 00011)
00959	9. RELEASE	(SEE LINE 00147)
00960	9. ISHIFT	
00961	9. EXITP	(SEE LINE 00033)
00962	8. GIMME	(SEE LINE 00153)
00963	8. EXITP	(SEE LINE 00033)
00964	7. PTREE	(SEE LINE 00957)
00965	7. RANDOM.	
00966	7. RELADO	
00967	8. ENTRYP	(SEE LINE 00011)
00968	8. GIMME	(SEE LINE 00153)
00969	8. PTRMRG	
00970	8. EXITP	(SEE LINE 00033)
00971	7. EXITP	(SEE LINE 00033)
00972	6. ITRAP	(SEE LINE 00028)
00973	6. AVAILBI.	
00974	7. RELADO	(SEE LINE 00966)
00975	7. RELEASE	(SEE LINE 00147)
00976	6. SCHEDUL	
00977	7. UNPACK	(SEE LINE 00189)
00978	7. RANDOM.	
00979	7. RENDEVU	
00980	8. UNPACK	(SEE LINE 00189)
00981	8. THXZY	(SEE LINE 00228)
00982	8. THY2HXL	(SEE LINE 00242)
00983	7. HEXADO	(SEE LINE 00252)
00984	7. GETHEX	(SEE LINE 00264)
00985	7. HEXDIST	
00986	7. ACFRAG	
00987	8. GIMME	(SEE LINE 00153)

00988	
00989	
00990	
00991	
00992	
00993	
00994	
00995	
00996	
00997	
00998	
00999	
01000	
01001	
01002	
01003	
01004	
01005	
01006	
01007	
01008	
01009	
01010	
01011	
01012	
01013	
01014	
01015	
01016	
01017	
01018	
01019	
01020	
01021	
01022	
01023	
01024	
01025	
01026	
01027	
01028	
01029	
01030	
01031	
01032	
01033	
01034	
01035	
01036	
01037	
01038	
01039	
01040	
01041	
01042	
	8. UNPACK (SEE LINE 00189)
	8. CRFLTML
	9. CREATE
	10. ENTRYP (SEE LINE 00011)
	10. GIMME (SEE LINE 00153)
	10. EXITP (SEE LINE 00033)
	9. GIMME (SEE LINE 00153)
	9. UNPACK (SEE LINE 00189)
	9. PACK (SEE LINE 00277)
	9. ADOBLOK (SEE LINE 00359)
	9. HISTORY (SEE LINE 00330)
	8. ADOBLOK (SEE LINE 00359)
	8. OPTPTH (SEE LINE 00607)
	8. RELEASE (SEE LINE 00147)
	8. PACK (SEE LINE 00277)
	8. FLTGEOM
	9. PACK (SEE LINE 00277)
	9. HEXCMZ (SEE LINE 00610)
	9. TH2PS (SEE LINE 00226)
	9. ATAN2.
	8. HEXOIST
	8. DELADD (SEE LINE 00162)
	6. DELADD (SEE LINE 00162)
	6. PLANOUT
	7. ENTRYP (SEE LINE 00011)
	7. OUTCI.
	7. EXITP (SEE LINE 00033)
	6. RLRAID
	7. ENTRYP (SEE LINE 00011)
	7. RLWAVE
	8. ENTRYP (SEE LINE 00011)
	8. RLGTYP
	9. ENTRYP (SEE LINE 00011)
	9. RELEASE (SEE LINE 00147)
	9. RLGTAX
	10. ENTRYP (SEE LINE 00011)
	10. RLFMAKT
	11. ENTRYP (SEE LINE 00011)
	11. RELEASE (SEE LINE 00147)
	11. EXITP (SEE LINE 00033)
	10. RELEASE (SEE LINE 00147)
	10. EXITP (SEE LINE 00033)
	9. EXITP (SEE LINE 00033)
	8. RELEASE (SEE LINE 00147)
	8. EXITP (SEE LINE 00033)
	7. RLCORD
	8. ENTRYP (SEE LINE 00011)
	8. RLWAVE
	9. ENTRYP (SEE LINE 00011)
	9. RELEASE (SEE LINE 00147)
	9. EXITP (SEE LINE 00033)
	8. RELEASE (SEE LINE 00147)
	8. EXITP (SEE LINE 00033)
	7. RELEASE (SEE LINE 00147)
	7. EXITP (SEE LINE 00033)

01043		
01044		
01045		
01046		
01047		
01048		
01049		
01050		
01051		
01052		
01053		
01054		
01055		
01056		
01057		
01058		
01059		
01060		
01061		
01062		
01063		
01064		
01065		
01066		
01067		
01068		
01069		
01070		
01071		
01072		
01073		
01074		
01075		
01076		
01077		
01078		
01079		
01080		
01081		
01082		
01083		
01084		
01085		
01086		
01087		
01088		
01089		
01090		
01091		
01092		
01093		
01094		
01095		
01096		
01097		
	6. EXITD	(SEE LINE 00033)
	5. EXITP	(SEE LINE 00033)
	4. POWOER	
	5. GETPTRS	(SEE LINE 00203)
	5. TFLYCRC	
	6. CRCTHMK	
	7. UNPACK	(SEE LINE 00189)
	7. RELEASE	(SEE LINE 00147)
	7. CRCKIL	(SEE LINE 00731)
	7. CRCTRK	(SEE LINE 00724)
	7. CRCLJSS	(SEE LINE 00413)
	7. AB2CRC	
	8. UNPACK	(SEE LINE 00189)
	8. RELEASE	(SEE LINE 00147)
	8. FINDBLK	(SEE LINE 00207)
	8. GIMME!	(SEE LINE 00153)
	8. ADOBLOK	(SEE LINE 00359)
	8. TAPES#	
	8. OUTCI.	
	8. DROPBLK	(SEE LINE 00299)
	7. INT2CRC	
	8. UNPACK	(SEE LINE 00189)
	8. FINDBLK	(SEE LINE 00207)
	8. RELEASE	(SEE LINE 00147)
	8. CRCKIL	(SEE LINE 00731)
	8. TAPES#	
	8. OUTCI.	
	8. DETECT	(SEE LINE 00702)
	8. MESBILO	(SEE LINE 00275)
	8. DELAJO	(SEE LINE 00162)
	8. DROPBLK	(SEE LINE 00299)
	8. GIMME!	(SEE LINE 00153)
	8. ADOBLOK	(SEE LINE 00359)
	7. BTN2CRC	
	8. UNPACK	(SEE LINE 00189)
	8. RELEASE	(SEE LINE 00147)
	8. TAPES#	
	8. OUTCI.	
	8. CRCLJSS	(SEE LINE 00413)
	8. CRCKIL	(SEE LINE 00731)
	8. CRCTRK	(SEE LINE 00724)
	6. UNSTAT	(SEE LINE 00309)
	6. AIRTHMK	
	7. HEXDIST	
	7. MESBILO	(SEE LINE 00275)
	7. DELAJO	(SEE LINE 00162)
	7. TGTWEX	(SEE LINE 00220)
	7. TAPES#	
	7. OUTCI.	
	7. RANDOM.	
	6. DOGTHMK	
	7. MESBILO	(SEE LINE 00275)
	7. DELAJO	(SEE LINE 00162)
	7. TAPES#	
	7. OUTCI.	

01098	7. GOTO43	(SEE LINE 00604)
01099	7. FUELCHK	(SEE LINE 00601)
01100	6. STATPAK	
01101	5. BOCTINK	
01102	6. BNPNSS	
01103	7. UNPACK	(SEE LINE 00189)
01104	7. GOTOER.	
01105	7. BNPN88	
01106	8. FILERUP	
01107	9. RANOOM.	
01108	9. TAPE6#	
01109	9. OJTCL.	
01110	9. GIMME	(SEE LINE 00153)
01111	9. YANK	(SEE LINE 00398)
01112	9. STICK	(SEE LINE 00432)
01113	9. DELADD	(SEE LINE 00162)
01114	8. UNPACK	(SEE LINE 00189)
01115	8. TVRANGE	
01116	9. THZPS	(SEE LINE 00226)
01117	9. ATAN2.	
01118	9. SORT.	
01119	9. COS.	
01120	9. TAPE6#	
01121	9. OJTCL.	
01122	9. AZILIM	
01123	10. SIN.	
01124	10. COS.	
01125	10. INSECT	
01126	11. TAN.	
01127	11. ATAN2.	
01128	10. SORT.	
01129	8. GIMME!	(SEE LINE 00153)
01130	8. DELADD	(SEE LINE 00162)
01131	8. SETASSN	
01132	9. UNPACK	(SEE LINE 00189)
01133	9. PACK	(SEE LINE 00277)
01134	9. STICK	(SEE LINE 00432)
01135	9. PRIORITY	
01136	9. RELOCAT	(SEE LINE 00437)
01137	9. DELADD	(SEE LINE 00162)
01138	8. SEEKENG	(SEE LINE 00854)
01139	8. PRIORITY	
01140	8. RELOCAT	(SEE LINE 00437)
01141	8. STICK!	(SEE LINE 00432)
01142	8. GOTOER.	
01143	8. DLYACT	(SEE LINE 00862)
01144	8. TAPE6#	
01145	8. OJTCL.	
01146	7. SEEKENG	(SEE LINE 00854)
01147	7. GIMME	(SEE LINE 00153)
01148	7. DLYACT	(SEE LINE 00862)
01149	7. BYUPDAT	
01150	8. UNPACK	(SEE LINE 00189)
01151	8. BNNOTRO	
01152	9. TAPE6#	

01153	9. OUTCI.	
01154	9. RELEASE	(SEE LINE 00147)
01155	9. DILOUT	(SEE LINE 00480)
01156	8. BATTJUT	
01157	9. YANK	(SEE LINE 00398)
01158	9. BNLALLE	(SEE LINE 00418)
01159	9. UNPACK	(SEE LINE 00189)
01160	9. CHKLAST	
01161	10. DROPP0S	(SEE LINE 00849)
01162	10. PRIORITY	
01163	10. RELOCAT	(SEE LINE 00437)
01164	10. DELADD	(SEE LINE 00162)
01165	9. GIMME	(SEE LINE 00153)
01166	9. RELEASE	(SEE LINE 00147)
01167	9. SSSL	(SEE LINE 00868)
01168	8. CDVAPLY	
01169	9. SEEKTAC	
01170	10. DROPP0S	(SEE LINE 00849)
01171	10. JNPACK	(SEE LINE 00189)
01172	10. GIMME	(SEE LINE 00153)
01173	10. PACK	(SEE LINE 00277)
01174	10. RELEASE	(SEE LINE 00147)
01175	10. SSSL	(SEE LINE 00868)
01176	10. DELADD	(SEE LINE 00162)
01177	10. ALLOBAT	(SEE LINE 00874)
01178	10. YANK	(SEE LINE 00398)
01179	10. READIL	(SEE LINE 00429)
01180	9. GIMME	(SEE LINE 00153)
01181	9. DLYACT	(SEE LINE 00862)
01182	9. RELEASE	(SEE LINE 00147)
01183	8. GIMME	(SEE LINE 00153)
01184	8. PACK	(SEE LINE 00277)
01185	8. SSSL	(SEE LINE 00868)
01186	8. DECRALD	
01187	9. MESSILD	(SEE LINE 00275)
01188	9. DELADD	(SEE LINE 00162)
01189	9. TAPE6#	
01190	9. OUTCI.	
01191	8. SEEKTAC	(SEE LINE 01169)
01192	8. DLYACT	(SEE LINE 00862)
01193	8. RELEASE	(SEE LINE 00147)
01194	6. BNPONER	
01195	7. GOTDER.	
01196	7. SKSBRK	
01197	8. UNPACK	(SEE LINE 00189)
01198	8. SEEKP	
01199	7. DROPP0S	(SEE LINE 00849)
01200	7. BNLALLE	(SEE LINE 00418)
01201	7. UNPACK	(SEE LINE 00189)
01202	7. CHKLAST	(SEE LINE 01160)
01203	7. SEEKTAC	(SEE LINE 01169)
01204	7. GIMME	(SEE LINE 00153)
01205	7. PACK	(SEE LINE 00277)
01206	7. DLYACT	(SEE LINE 00862)
01207	7. SEEKENG	(SEE LINE 00854)

01208	7. MESBILD	(SEE LINE 00275)
01209	7. DELA00	(SEE LINE 00162)
01210	7. YANK	(SEE LINE 00398)
01211	7. STICK	(SEE LINE 00432)
01212	7. READIL	(SEE LINE 00429)
01213		
01214	6. BNCMOPR	
01215	7. GOTOER.	
01216	7. ACCEPT	
01217	8. UNPACK	(SEE LINE 00189)
01218	8. SEEKP	
01219	8. GIMME!	(SEE LINE 00153)
01220	8. STOP.	
01221	8. STICK	(SEE LINE 00432)
01222	8. DETECT	(SEE LINE 00702)
01223	8. DELA00	(SEE LINE 00162)
01224	8. PACK	(SEE LINE 00277)
01225	8. TAPES#	
01226	8. OJTCL.	
01227	8. BYALCOV	
01228	9. UNPACK	(SEE LINE 00189)
01229	9. PACK	(SEE LINE 00277)
01230	9. GIMME	(SEE LINE 00153)
01231	9. SLL	(SEE LINE 00868)
01232	9. CANCEL0	(SEE LINE 00424)
01233	9. SEEXTFJ	(SEE LINE 00443)
01234	9. RELEASE	(SEE LINE 00147)
01235	9. PATDEC	
01236	10. CANCEL0	(SEE LINE 00424)
01237	10. SEEXTFJ	(SEE LINE 00443)
01238	9. BATT0OV	
01239	10. TRKCHK	(SEE LINE 00444)
01240	10. ALLOFU	(SEE LINE 00462)
01241	10. ALLOPAT	(SEE LINE 00468)
01242	9. OLYACT	(SEE LINE 00862)
01243	9. TAPE6#	
01244	9. OJTCL.	
01245	8. RELEASE	(SEE LINE 00147)
01246	8. BYMEDUP	
01247	9. SEEKP	
01248	9. UNPACK	(SEE LINE 00189)
01249	9. DELA00	(SEE LINE 00162)
01250	9. RELOCAT	(SEE LINE 00437)
01251	9. BATT0OV	(SEE LINE 01237)
01252	9. MESBILD	(SEE LINE 00275)
01253	9. RELEASE	(SEE LINE 00147)
01254	8. OJTA	(SEE LINE 00403)
01255	7. MESBILD	(SEE LINE 00275)
01256	7. DELA00	(SEE LINE 00162)
01257	7. SEEKP	
01258	7. DROPP0S	(SEE LINE 00849)
01259	7. OILOJT	(SEE LINE 00480)
01260	6. BNPONFA	
01261	7. SEEKP	
01262	7. TAPES#	
	7. OUTCL.	

01263	
01264	
01265	
01266	
01267	
01268	
01269	
01270	
01271	
01272	
01273	
01274	
01275	
01276	
01277	
01278	
01279	
01280	
01281	
01282	
01283	
01284	
01285	
01286	
01287	
01288	
01289	
01290	
01291	
01292	
01293	
01294	
01295	
01296	
01297	
01298	
01299	
01300	
01301	
01302	
01303	
01304	
01305	
01306	
01307	
01308	
01309	
01310	
01311	
01312	
01313	
01314	
01315	
01316	
01317	
	7. BNRECDV
	8. UNPACK (SEE LINE 00189)
	8. WITHDRAW
	9. UNPACK (SEE LINE 00189)
	9. GIMME (SEE LINE 00153)
	9. PACK (SEE LINE 00277)
	9. SELL (SEE LINE 00868)
	9. DECALC (SEE LINE 01186)
	9. SEEKENG (SEE LINE 00854)
	9. OLYACT (SEE LINE 00862)
	9. RELEASE (SEE LINE 00147)
	8. PACK (SEE LINE 00277)
	6. BNPNFD
	7. SEEK#
	7. TAPES#
	7. OUTCI.
	7. RELEASE (SEE LINE 00147)
	7. YANK (SEE LINE 00398)
	7. DROPPQS (SEE LINE 00849)
	7. OILOUT (SEE LINE 00480)
	7. MESBILD (SEE LINE 00275)
	7. DELADD (SEE LINE 00162)
	6. RELEASE (SEE LINE 00147)
	6. SDIGEST
	7. UNPACK (SEE LINE 00189)
	7. GIMME (SEE LINE 00153)
	7. PACK (SEE LINE 00277)
	7. TAPES#
	7. OUTCI.
	7. RANDOM.
	7. BNNWTRK
	8. CHKCDV
	9. UNPACK (SEE LINE 00189)
	9. INRANGE (SEE LINE 01115)
	9. GIMME (SEE LINE 00153)
	9. TAPES#
	9. OUTCI.
	9. OJTA (SEE LINE 00403)
	9. DELADD (SEE LINE 00162)
	8. RVLALLE (SEE LINE 00418)
	8. SETASSN (SEE LINE 01131)
	8. SEEKTAC (SEE LINE 01169)
	8. GIMME (SEE LINE 00153)
	8. OLYACT (SEE LINE 00862)
	7. BNNWTRK
	8. MESBILD (SEE LINE 00275)
	8. DELADD (SEE LINE 00162)
	8. RVLALLE (SEE LINE 00418)
	8. INRANGE (SEE LINE 01115)
	8. PREPAFU
	9. UNPACK (SEE LINE 00189)
	9. PACK (SEE LINE 00277)
	9. AJTOPRI
	9. STICK (SEE LINE 00432)
	9. RELOCAT (SEE LINE 00437)

01318	9. DELADD	(SEE LINE 00162)
01319	8. BATTMOV	(SEE LINE 01237)
01320	8. GIMME	(SEE LINE 00153)
01321	8. DLYACT	(SEE LINE 00862)
01322	8. STICK	(SEE LINE 00432)
01323	8. RELOCAT	(SEE LINE 00437)
01324	7. YANK	(SEE LINE 00398)
01325	7. RELEASE	(SEE LINE 00147)
01326	7. TOADIL	(SEE LINE 00454)
01327	7. STICK	(SEE LINE 00432)
01328	7. BNCONTC	
01329	8. CHKCOV	(SEE LINE 01294)
01330	8. DROPPOS	(SEE LINE 00849)
01331	8. RNLALLE	(SEE LINE 00419)
01332	8. SETASSN	(SEE LINE 01131)
01333	8. SEEKTAC	(SEE LINE 01169)
01334	8. GIMME	(SEE LINE 00153)
01335	8. DLYACT	(SEE LINE 00862)
01336	8. CHKLAST	(SEE LINE 01150)
01337	8. BYCONVD	
01338	9. TAPE6#	
01339	9. OUTCI.	
01340	9. GIMME	(SEE LINE 00153)
01341	9. MESBILO	(SEE LINE 00275)
01342	9. PRIORTY	
01343	9. PACK	(SEE LINE 00277)
01344	9. DELADD	(SEE LINE 00162)
01345	9. YANK	(SEE LINE 00398)
01346	9. STICK	(SEE LINE 00432)
01347	9. RELEASE	(SEE LINE 00147)
01348	9. DROPPOS	(SEE LINE 00849)
01349	9. SEEKTAC	(SEE LINE 01169)
01350	9. DLYACT	(SEE LINE 00862)
01351	8. BNRECOV	(SEE LINE 01263)
01352	8. TAPES#	
01353	8. OUTCI.	
01354	8. YANK	(SEE LINE 00398)
01355	8. STICK	(SEE LINE 00432)
01356	8. TOADIL	(SEE LINE 00454)
01357	8. DILDJT	(SEE LINE 00480)
01358	7. BYCONTC	
01359	8. MESBILO	(SEE LINE 00275)
01360	8. DELADD	(SEE LINE 00162)
01361	8. YNRANGE	(SEE LINE 01115)
01362	8. BATCEAS	(SEE LINE 00423)
01363	8. RNLALLE	(SEE LINE 00418)
01364	8. PREPAFU	(SEE LINE 01312)
01365	8. BATTMOV	(SEE LINE 01237)
01366	8. GIMME	(SEE LINE 00153)
01367	8. DLYACT	(SEE LINE 00862)
01368	8. BYCONVD	
01369	9. TAPE6#	
01370	9. OUTCI.	
01371	9. DELADD	(SEE LINE 00162)
01372	9. AJTOPRI	

01373	
01374	9. STICK (SEE LINE 00432)
01375	9. RELOCAT (SEE LINE 00437)
01376	9. BATTGOV (SEE LINE 01237)
01377	9. GIMME (SEE LINE 00153)
01378	9. OLYACT (SEE LINE 00862)
01379	8. UNPACK (SEE LINE 00189)
01380	8. BYALCOV (SEE LINE 01226)
01381	8. STICK (SEE LINE 00432)
01382	8. RELOCAT (SEE LINE 00437)
01383	8. YANK (SEE LINE 00398)
01384	8. TJOADLI (SEE LINE 00454)
01385	7. OLYACT (SEE LINE 00862)
01386	7. BNCOVLS
01387	8. DROPPOS (SEE LINE 00849)
01388	8. DILOUT (SEE LINE 00480)
01389	8. MESBILD (SEE LINE 00275)
01390	8. DELAOD (SEE LINE 00162)
01391	7. BYCOVLS
01392	8. RATCEAS (SEE LINE 00423)
01393	8. BNLALLE (SEE LINE 00418)
01394	8. MESBILD (SEE LINE 00275)
01395	8. DELAOD (SEE LINE 00162)
01396	8. DILOUT (SEE LINE 00480)
01397	7. BNLALLE (SEE LINE 00418)
01398	7. DELAOD (SEE LINE 00162)
01399	6. BNPNDA
01400	7. YANK (SEE LINE 00398)
01401	7. UNPACK (SEE LINE 00189)
01402	7. GOTOER.
01403	7. SETASSN (SEE LINE 01131)
01404	7. SEEKTAC (SEE LINE 01169)
01405	7. CHKCOV (SEE LINE 01294)
01406	7. BNLALLE (SEE LINE 00418)
01407	7. GIMME (SEE LINE 00153)
01408	7. OLYACT (SEE LINE 00862)
01409	7. BNCONTC (SEE LINE 01328)
01410	7. SEEKENG (SEE LINE 00854)
01411	7. RELEASE (SEE LINE 00147)
01412	6. TAPE6#
01413	6. OUTCI.
01414	6. DROPPOS (SEE LINE 00849)
01415	6. BNLALLE (SEE LINE 00418)
01416	6. MESBILD (SEE LINE 00275)
01417	6. DELAOD (SEE LINE 00162)
01418	6. DILOUT (SEE LINE 00480)
01419	6. BNPNBD
01420	7. TAPES#
01421	7. OUTCI.
01422	7. BNNOTRO (SEE LINE 01151)
01423	7. BATTOUT (SEE LINE 01156)
01424	7. COVAPLY (SEE LINE 01168)
01425	7. RELEASE (SEE LINE 00147)
01426	6. SAMATON
01427	7. UNPACK (SEE LINE 00189)
	7. AUTOPRI

01428	7. RELOCAT	(SEE LINE 00437)
01429	7. RELEASE	(SEE LINE 00147)
01430	7. DILOJT	(SEE LINE 00480)
01431	7. TOADIL	(SEE LINE 00454)
01432	7. DELADD	(SEE LINE 00162)
01433	7. TAPES#	
01434	7. OUTCI.	
01435	6. ALLOBAT	(SEE LINE 00874)
01436	5. BTRYTNK	
01437	6. BYCMOPR	
01438	7. ACCEPT	(SEE LINE 01215)
01439	7. SEEKP	
01440	7. TAPES#	
01441	7. OUTCI.	
01442	7. BATCEAS	(SEE LINE 00423)
01443	7. DILOJT	(SEE LINE 00480)
01444	7. BYALCOV	(SEE LINE 01226)
01445	7. BYHEDJP	(SEE LINE 01245)
01446	7. RELEASE	(SEE LINE 00147)
01447	6. SDIGEST	(SEE LINE 01286)
01448	6. SEEKP	
01449	6. BYPONTM	
01450	7. GIMME	(SEE LINE 00153)
01451	7. UNPACK	(SEE LINE 00189)
01452	7. PACK	(SEE LINE 00277)
01453	7. TAPES#	
01454	7. OUTCI.	
01455	7. BYCONTC	(SEE LINE 01358)
01456	7. TRYSDJT	(SEE LINE 00547)
01457	7. DELADD	(SEE LINE 00162)
01458	7. RELEASE	(SEE LINE 00147)
01459	6. BYPONER	
01460	7. UNPACK	(SEE LINE 00189)
01461	7. CANCEL0	(SEE LINE 00424)
01462	7. SEEKTFU	(SEE LINE 00443)
01463	7. BATTCOV	(SEE LINE 01237)
01464	7. DILOJT	(SEE LINE 00480)
01465	7. PACK	(SEE LINE 00277)
01466	7. DELADD	(SEE LINE 00162)
01467	7. PTPONER	
01468	8. UNPACK	(SEE LINE 00189)
01469	8. PACK	(SEE LINE 00277)
01470	8. CANCEL0	(SEE LINE 00424)
01471	8. SEEKTFU	(SEE LINE 00443)
01472	8. DILOJT	(SEE LINE 00480)
01473	8. DELADD	(SEE LINE 00162)
01474	7. MESBILD	(SEE LINE 00275)
01475	7. RELEASE	(SEE LINE 00147)
01476	6. TAPE#	
01477	6. OUTCI.	
01478	6. YANK	(SEE LINE 00398)
01479	6. BATCEAS	(SEE LINE 00423)
01480	6. BNLALLE	(SEE LINE 00418)
01481	6. MESBILD	(SEE LINE 00275)
01482	6. DELADD	(SEE LINE 00162)

01483	
01484	
01485	
01486	
01487	
01488	
01489	
01490	
01491	
01492	
01493	
01494	
01495	
01496	
01497	
01498	
01499	
01500	
01501	
01502	
01503	
01504	
01505	
01506	
01507	
01508	
01509	
01510	
01511	
01512	
01513	
01514	
01515	
01516	
01517	
01518	
01519	
01520	
01521	
01522	
01523	
01524	
01525	
01526	
01527	
01528	
01529	
01530	
01531	
01532	
01533	
01534	
01535	
01536	
01537	
	6. BYPNFD
	7. SEEK
	7. YANK (SEE LINE 00398)
	7. RELEASE (SEE LINE 00147)
	7. BATCEAS (SEE LINE 00423)
	7. OILOJT (SEE LINE 00480)
	7. BYENDPS (SEE LINE 00395)
	7. MESBILD (SEE LINE 00275)
	7. DELADD (SEE LINE 00162)
	6. SAMATN (SEE LINE 01425)
	6. BYPNRL
	7. UNPACK (SEE LINE 00189)
	7. GOTOER.
	7. FILERUP (SEE LINE 01106)
	7. INRANGE (SEE LINE 01115)
	7. PREPAFU (SEE LINE 01312)
	7. MESBILD (SEE LINE 00275)
	7. DELADD (SEE LINE 00162)
	7. SEEKTFU (SEE LINE 00443)
	7. TAPES#
	7. OUTCI.
	6. BYPNRS
	7. UNPACK (SEE LINE 00189)
	7. RELOAD
	8. GIMME (SEE LINE 00153)
	8. GOTOER.
	8. DELADD (SEE LINE 00162)
	8. RESUPPLY
	9. GIMME (SEE LINE 00153)
	9. DELADD (SEE LINE 00162)
	7. TAPES#
	7. OUTCI.
	7. RELEASE (SEE LINE 00147)
	4. TOWER
	5. GETPTRS (SEE LINE 00203)
	5. TAPES#
	5. OUTCI.
	5. UNPACK (SEE LINE 00189)
	5. UOLLOAD (SEE LINE 00596)
	5. INITACQ
	6. ENTRYP (SEE LINE 00011)
	6. GIMME (SEE LINE 00153)
	6. ADDBLOK (SEE LINE 00359)
	6. ALOG.
	6. MESSAGE (SEE LINE 00012)
	6. RITEI (SEE LINE 00015)
	6. XTOI.
	6. NOWUCIT (SEE LINE 00350)
	6. EXITP (SEE LINE 00033)
	5. DELADD (SEE LINE 00162)
	5. PACK (SEE LINE 00277)
	5. FINDBLK (SEE LINE 00207)
	5. REDEBRF
	6. UNPACK (SEE LINE 00189)
	6. FINDBLK (SEE LINE 00207)

01538		6. WIPEOUT	
01539		6. GIMME	(SEE LINE 00153)
01540		6. ADDBLOK	(SEE LINE 00359)
01541		6. PELADD	(SEE LINE 00966)
01542		6. DESTROY	(SEE LINE 00343)
01543		5. GIMME	(SEE LINE 00153)
01544	5.	GOGETEM	
01545		6. UNPACK	(SEE LINE 00189)
01546		6. FINDBLK	(SEE LINE 00207)
01547		6. CRFLTML	(SEE LINE 00989)
01548		6. TAPE6#	
01549		6. OUTCI.	
01550		6. GIMME	(SEE LINE 00153)
01551		6. ADDBLOK	(SEE LINE 00359)
01552		6. PTRAND	
01553		7. UNPACK	(SEE LINE 00189)
01554		7. RANDOM.	
01555		7. HEXADD	(SEE LINE 00252)
01556		7. HEXMJLT	(SEE LINE 00353)
01557		7. GETHEX	(SEE LINE 00264)
01558		6. PACK	(SEE LINE 00277)
01559		6. FLTGEOM	(SEE LINE 01003)
01560		6. UOLLOAD	(SEE LINE 00596)
01561		6. INITACO	(SEE LINE 01522)
01562		6. DELADD	(SEE LINE 00162)
01563		6. MESBILD	(SEE LINE 00275)
01564	5.	MESBILD	(SEE LINE 00275)
01565	5.	DESTROY	(SEE LINE 00343)
01566	4.	UMPIRE	
01567		5. UNPACK	(SEE LINE 00189)
01568		5. THM2PS	(SEE LINE 00226)
01569		5. SORT.	
01570		5. XTOI.	
01571		5. HISTORY	(SEE LINE 00330)
01572		5. RANDOM.	
01573		5. TAPE6#	
01574		5. OUTCI.	
01575		5. GIMME	(SEE LINE 00153)
01576		5. PACK	(SEE LINE 00277)
01577		5. DELADD	(SEE LINE 00162)
01578		5. DESTROY	(SEE LINE 00343)
01579		5. RELEASE	(SEE LINE 00147)
01580		5. NUKBLND	(SEE LINE 00509)
01581	4.	TAPE6#	
01582	4.	OUTCI.	
01583	4.	OUTCR.	
01584	4.	EXITP	(SEE LINE 00033)
01585	3.	SNAP	(SEE LINE 00168)
01586	3.	UNSNAP	(SEE LINE 00196)
01587	3.	RELEASE	(SEE LINE 00147)
01588	3.	SECONO	
01589	3.	HLTPNT	
01590		4. ENTRYP	(SEE LINE 00011)
01591		4. SECONO	
01592		4. HALT	(SEE LINE 00009)

01593
01594
01595
01596

4. EXITP (SEE LINE 00033)
3. EXITP (SEE LINE 00033)
2. EXITP (SEE LINE 00033)
2. END.

PAGE 100

3. Post Processor

LIST OF SUBROUTINES - RECORD

PAGE 1

1. EOF
2. INDEX
3. MESSAGE
4. PAGE
5. RECORDU
6. RITET
7. RITER
8. TABOUT

LIST OF FORTRAN LIBRARY ROUTINES - RECORD

1. INPCT.
2. INPFI.
3. OUTCT.
4. OUTCR.
5. QINTRY.
6. STOP.
7. TAPE6#

SUBROUTINE REFERENCE LIST - RECORD

PAGE 3

1. EOF		CALLLED BY:	RECORD
2. INDEX	CALLS:	CALLLED BY:	RECORD
	TAPE6#		
	OUTCI.		
3. MESSAGE	CALLS:	CALLLED BY:	TABOUT
	TAPE6#		RECORD
	OUTCI.		
4. PAGE	CALLS:	CALLLED BY:	TABOUT
	TAPE6#		RECORD
	OUTCI.		
5. RECORD	CALLS:		
	QINTRY.		
	INPCI.		
	INPFI.		
	PAGE		
	OUTCI.		
	EOF		
	INDEX		
	MESSAGE		
	RITEI		
	RITER		
	TABOUT		
	STOP.		
6. RITEI	CALLS:	CALLLED BY:	RECORD
	TAPE6#		
	OUTCI.		
7. RITER	CALLS:	CALLLED BY:	RECORD
	TAPE6#		
	OUTCI.		
8. TABOUT	CALLS:	CALLLED BY:	RECORD
	PAGE		
	MESSAGE		
	TAPE6#		
	OUTCI.		
	OUTCR.		

FORTRAN LIBRARY REFERENCE LIST - RECORD

PAGE 6

1. INPCI.	CALLED BY: RECORD
2. INPFI.	CALLED BY: RECORD
3. OUTCI.	CALLED BY: TABOUT RITER RITEI PAGE MESSAGE INDEX RECORD
4. OUTCR.	CALLED BY: TABOUT
5. JINTRY.	CALLED BY: RECORD
6. STOP.	CALLED BY: RECORD
7. TAPE6*	CALLED BY: TABOUT RITER RITEI PAGE MESSAGE INDEX

RECORD SUBROUTINE CALLING HIERARCHY -

PAGE 5

00001	1. RECORD	
00002	2. QINTRY.	
00003	2. INPCI.	
00004	2. INPFI.	
00005	2. PAGE	
00006	3. TAPE6#	
00007	3. OUTCI.	
00008	2. OUTCI.	
00009	2. EOF	
00010	2. INDEX	
00011	3. TAPE6#	
00012	3. OUTCI.	
00013	2. MESSAGE	
00014	3. TAPE6#	
00015	3. OUTCI.	
00016	2. RITEI	
00017	3. TAPE6#	
00018	3. OUTCI.	
00019	2. RITER	
00020	3. TAPE6#	
00021	3. OUTCI.	
00022	2. TABOUT	
00023	3. PAGE	(SEE LINE 00005)
00024	3. MESSAGE	(SEE LINE 00013)
00025	3. TAPE6#	
00026	3. OUTCI.	
00027	3. OUTCR.	
00028	2. STOP.	

APPENDIX I

MADEM DEBUG ROUTINES

This appendix contains an alphabetical list and description of the debug routines available in MADEM. Following this list is a more detailed description of how to implement certain of these capabilities.

- ADUMP - Dumps to the printer any array in octal.
Parameters:
IARRAY - array to be dumped
ISTART - offset of first word to be dumped
LENGTH - number of words to be dumped
INAME - hollerith name of the array
- ANALYZE - A separate program that analyzes a binary file and gives the following information:
Cross reference of all routine calls
Complete calling hierarchy
- BLOCK
DATA
ROUTINES - 41 routines, each representing a data block, that can be manipulated to create data structure display routines. Each block routine is capable of printing data blocks or a list of data blocks, and of calling other block routines to print subordinate lists.
Parameters:
POINTER - pointer to first data block
LEVEL - number of hierarchical types of blocks to print in the data structure
MAXBLKS - number of blocks in the main list to be printed.
- CLIST
(and
CLIST2) - Common block dump routine. Prints key common pointer values.

- DBGREAD - Debug parameter read routine. Reads all input parameters used for debug purposes. These parameters are interpreted and the necessary flags are set to activate these parameters. Invalid parameters are ignored.
- DISPLAT - Display DATFILE data structure. Uses 13 of the block data routines. Activated by a debug parameter.
- ENTRYD - Entry debug routine. Called at the beginning of most MADEM subroutines. Keeps track of the calling hierarchy in the pushdown stack and of the last 50 routines called in the circular list. Counts the number of times each routine calls ENTRYD and, along with EXITP, times the execution of these routines. Optionally, ENTRYD can call debug routines ITRAP and ICHEC. Also, can optionally print specified routine trace messages. These options are set through debug parameters.

Parameter:

SEGNUM - segment number of calling routine

- ENTSTAT - Prints vector of routine entry counts, execution times, trace message flags, and debug call flags.

- EXITP - Same as ENTRYD, but called at the end of a routine rather than at the beginning.

Parameter:

SEGNUM - segment number of calling routine

- HALT - Used whenever the simulation is to be stopped. Performs the following functions, mostly through subroutine calls:

- creates hold files for restarts
- prints name of calling subroutine
- prints reason for termination.
- prints pushdown stack
- prints names of last 50 routines called
- calls ENTSTAT (see ENTSTAT)

- calls CLIST (see CLIST)
- optionally (controlled by debug parameter) prints ISPACE.
- Stops the simulation.

Parameters:

SEGNUM - segment number of calling routine

MSGHALT - forty character message

ICHECK - Checks specified locations in ISPACE, and prints a message when the value of that location changes. The message indicates the old value, the new value, and the last non-debug routine called. The ISPACE locations are selected through debug parameters read by DBGREAD. DEBUG must be set to "ON" for ICHECK to be called from any given routine (see DBGREAD).

Parameter:

SEGNUM - segment number of calling routine

ICOMP - A separate program that compares two sets of hold files and indicates when there are differences in the ISPACES. Used when midasizing, to insure the midasized version runs the same as the unmidasized version.

Parameters:

ISTART - first word of ISPACE to be compared

IMAX - last word of ISPACE to be compared

MAXDIFF - maximum number of mismatches before stopping

IEXTRA - maximum number of extra words printed if one ISPACE is larger than the other.

HSIZE1 - number of ISPACE words in each hold file for second ISPACE.

- ISDUMP - Dumps to printer selected portions or all of the ISPACE array. Will always print at least the first ten words.
Parameters:
 ISTART - first word to be dumped
 LENGTH - number of words to be dumped
- ITRAP - Checks ISPACE of zero and ISPACE locations one through eight for proper values. ISPACE of zero should always be equal to 99999999.0, and other eight locations should always be zero. If any of these locations have improper values, then HALT is called to stop the simulation and print debug information. DEBUG must be set to "ON" for ITRAP to be called from ENTRYYP and EXITP (see DBGRAD).
Parameter:
 SEGNUM - segment number of calling routine, or of routine that called ENTRYYP/EXIT.
- LOCATE - A batch text search program that can be used as a MIDAS cross reference program.
- NIPULSTOR - A post run debugging facility that can:
 1. Print selected areas of ISPACE.
 2. Dump the C2 data structures.
 3. Dump the EVENT tree.
 4. Dump structures of a given unit.
 5. Call CLIST.
See the detailed description at the end of this appendix.
- RECCON - Activated only by RECOUR in the event of abnormal job termination and subsequent recovery. Calls HALT to print debug information and stop the simulation.
- RECER - Prints current calling hierarchy (pushdown stack) and the names of the last 50 subroutines called (circular list).

RECOUR - Allows the MADDEM program to regain control of execution at the time that abnormal job termination would otherwise occur. RECOUR calls RECCON in the event of catastrophic failure. RECOUR is automatically initialized at the beginning of the MADDEM Program. RECOUR may be turned off by using debug parameter "RECOUR=OFF."

Parameters:

NAME - name of the routine to be executed if flagged conditions occur (RECCON)

FLAGS - octal value of error conditions that trip RECOUR (077)

CHECKSUM - No checksum desired (0)

ROUTER - Same as RECER, but with SEGNUM as a parameter, so the calling routine name can be printed.

MADDEM DEBUG PARAMETERS

The INPUT file to MADDEM holds various parameters that affect only the particular volume that is being run. The first input card holds seven numbers described under MADDEM Operations. This card is mandatory. Debug parameters follow this first card, and are entirely optional. There are a variety of debug options which may be turned on or off by using debug parameters. These options and their corresponding parameters are listed below. Parameters, except where noted otherwise, must begin in column one. All parameters are actually ten characters long, with either leading or trailing blanks implied.

The debug options:

1. Debug Status

For each routine that calls ENTRYR and EXITR, the debug status is set to either "ON" or "OFF". When debug is "ON" for a routine, then ENTRYR and EXITR have the addition of calling ITRAP and ICHECK when processing

that routine. This slows down execution considerably when many routines are "ON," but is a valuable debug tool.

DEFAULT: Debug is "OFF" for all routines.

PARAMETERS:

"DEBUG=ON~~BB~~" - turns delay to "ON" for all routines

"DXXXXXXXX~~BB~~" - where XXXXXXXX is a routine name.

Changes the debug status of that routine only. If it was "ON," it is set to "OFF," and vice-versa. By using "DEBUG=ON~~BB~~" followed by a few occurrences of this parameter, all routines but a few can be set to "ON." Likewise, by only using this parameter, only a few routines can be set to "ON."

2. Trace Status

The same as debug status, except that what is being turned "ON" and "OFF" is the printing of subroutine call trace messages.

DEFAULT: Trace is "OFF" for all routines.

PARAMETERS:

"TRACEON~~BB~~" - turns on trace messages for all routines.

"TXXXXXXXX~~BB~~" - same as "DXXXXXXXX~~BB~~" but for the trace status.

Never use "TRACE=ON" by itself. The resulting output will be thousands of pages of trace messages.

3. Recovery Status

Controls the initializing of system recover routine RECOUR. When recover is "ON," then RECOUR will be initiated upon abnormal termination. When Recover is "OFF," no calls will be made to RECOURI.

DEFAULT: Recover is set to "ON"

PARAMETER:

"RECOUR=OFF" - turns off recovery routine.

4. Icheck Status

Controls ISPACE locations checked by debug routine ICHECK. To have ISPACE locations checked, the word "CHECK" must start in column 1, followed on the same card by up to seven decimal ISPACE pointer right justified ending in column $10x$, where $x = 2$ thru 8. All check cards together may not have more than ten ISPACE locations.

DEFAULT: no ISPACE locations are checked.

5. Release Status

The allocation of blocks in ISPACE is controlled by two routines: GIMME and RELEASE. Release un-allocates previously used storage for future use. Sometimes it is advantageous to turn off release so that all new blocks will be allocated at the end of "used" ISPACE (free space). This is done by turning the release status off. It also must be indicated after which event within the volume that release is turned off. To turn release off, put "RELEASE~~000~~" beginning in column one, followed on the same card by a decimal number ending in column 20. This number is the event after which release is turned off.

DEFAULT: Release is never turned off.

6. Stop Status (Pre-processor only)

The Stop Status controls how far the preprocessor runs before stopping. There is no restart capability for the preprocessor; the early stops are for debugging purposes only.

DEFAULT: Pre-processor runs to normal completion.

PARAMETERS:

"STOP=ODAT~~0~~" - stops after reading DATFILE.

"STOP=UOIL~~0~~" - stops after reading UOIL.

"STOP=DEL" - stops after planning first event.

7. Datfile Display (Pre-processor only)

To get DISPDAT to print the DATFILE data structure, use parameter "DATFLE=ON~~0~~".

DEFAULT: Datfile is not displayed

PARAMETER:

"DATFILE=ONØ"

8. Ispace Dump

DEFAULT: only first ten words of ISPACE are dumped

PARAMETER:

"DUMP=ONØØØ" - all of ISPACE dumped.

NIPUL8TOR

NIPUL8TOR is a fortran program that was written to use as a debugging tool for MADEM. NIPUL8TOR can dump selected areas of ISPACE, selected data structures, or MADEM's common blocks.

At the end of each MADEM run (or volume of a run), MADEM saves ISPACE and all other common blocks in a series of files. NIPUL8TOR gets ISPACE and the common blocks from this series of files. To maintain compatibility, NIPUL8TOR uses MADEM's fetch routine to retrieve the data from these files. NIPUL8TOR also uses MADEM's CLIST subroutine, as well as the subroutines that CLIST calls.

USING NIPUL8TOR

To use NIPUL8TOR, you need:

- 1) The binary file "NIPUL8TOR"
- 2) The series of MADEM files that holds ISPACE and the common blocks
- 3) The correct set of NIPUL8TOR directives that tell NIPUL8TOR exactly which dumps you want.

The sample JCL deck on the next page shows the input that will give you an end of Volume 3 dump that exercises all the NIPUL8TOR options. The first four attach commands access the MADEM files that were dumped by a volume 3 run. The NIPUL8TOR directives (commands) and directive parameters are also shown.

DIRECTIVES FOR NIPUL8TOR

Directives, submitted through an input deck, tell NIPUL8TOR exactly what to dump. Depending on the directive, there may be zero, one, or two parameters on the input card following the directive. The directive appears alone on a card. The directive is a digit (1-7), and directive parameters are either decimal integers or commands, as indicated below. There is no limit to the number of directives used in the SIPUL8TOR run, or to the number of repetitions of any one directive. The directives are not order dependent.

DIRECTIVE 1: Dump a selected area of ISPACE. This directive will dump a chunk of ISPACE, as defined by the two parameters.

PARAMETERS

- 1) First word of ISPACE to be dumped.
- 2) Last word of ISPACE to be dumped.

Obviously, the first PARM must be less than or equal to the second PARM.

DIRECTIVE 2: Dump around a word of ISAPCE. This directive will also dump a chunk of ISPACE as defined by the two parameters, but the PARMS have different meanings.

PARAMETERS

- 1) Middle word of ISPACE area to be dumped.
- 2) Number of words on either side of the middle word, to be dumped.

DIRECTIVE 3: Dump the C2 structure CTREED. This directive will display a C2 tree as defined by the parameter. It may be used to dump the red or blue C2 trees, or any subset of a C2 tree. For each unit in the C2 tree, the SB, SDB, C2, and unit status board blocks will be displayed.

PARAMETERS

3 options that define the C2 tree:

- 1) RED - dumps the red C2 tree
- 2) BLUE - dumps the blue C2 tree
- 3) PTR=IIIIII, where IIIIII = pointer to buffer at top of C2 tree. (IIIIII is a right justified 6 digit integer).

To dump a subset of a tree, use PTR=XXXXXX, where XXXXXX = PTRC2 + 1, where PTRC2 points to unit above the top unit in the subset. (i.e. PJRCZ + 1 simulates the buffer).

DIRECTIVE 4: Dump leftist tree and event node info. This directive will dump the leftist tree and corresponding event blocks.

PARAMETERS

One PARM, defines leftist tree.

2 options:

- 1) DEL - dumps discrete event list
- 2) PTR = IIIIII, where IIIIII is a right justified 6 digit integer that points to the top node in the tree.

DIRECTIVE 5: Not used. Future plans call for a hex structure dump.

DIRECTIVE 6: Dump structures of a given unit. This directive can be used to dump the blocks of a given unit, where the unit is displayed as in the C2 dump.

PARAMETER

Decimal integer pointer to the C2 unit to be dumped.

DIRECTIVE 7: Dump the common blocks. This directive will display most of MADEM's common blocks.

No PARAMETERS, do not use a second card.

APPENDIX J
 MADEM DATA STRUCTURE CROSS REFERENCE

MADEM	ABOUEUF	ABSLF	ABVSCOR	ABZCRC	ACCEPT	ACFRAG	ADASASS	ADDBLOK	AIRTHNK	ALTUBAT	ALLOFU	ALLOPAT	AMMOCHK	ASSIGN	ATRASES	ATTACK	AUTOPRI	AVAILBI	AZIMM	BADMOVE	BATGEAS	BATTCOV	BATIOUT
APPK																							
ACFRAG																							
AFM																							
AGPD																							
AGPK																							
CLOCK																							
COMOUT																							
COM PTR																							
COM SCS																							
CRCCOM																							
CSTK																							
DATA																							
DEBUG																							
DFLAGS																							
XTRACE																							
STEXT																							
FS																							
HALT																							
INITPR																							
IODEV																							
JOPT																							
LIMITS																							
MASK																							
MODVAR																							
MXMIS																							
PATH																							
SAMPK																							
SAMPTRS																							
SEMINFO																							
SPACE																							
SPSTAT																							
STATBD																							
THRPLN																							
TRACK																							
TYPES																							
CHOLD																							

Feature	BYALCOV	BYCMDPR	BYCONHD	BYCONLS	BYCONIC	BYENDPS	BYHEDUP	BYNOTRD	BYNWTRK	BYPASUP	BYPONER	BYPONED	BYPONHI	BYPONRS	BYPONTM	BYKCHK	BYUPDAT	CANCAIO	CANDTGT	CENTER	ESYCRG	CHKCOV	CHKLASE	CUST
APPK																								
ACFRAG																								
AFM																								
AGPD																								
AGPK																								
CLOCK																								
COMOUT																								
COM PTR																								
COM SCS																								
CRCOM																								
CSTK																								
DATA																								
DEBUG																								
DFLAGS																								
XTRACE																								
STEXT																								
FS																								
HALT																								
INPTR																								
IDEV																								
JOPT																								
LIMITS																								
MASK																								
MODVAR																								
MXMIS																								
PATH																								
SAMPK																								
SAMPTRS																								
SEMINFO																								
SPACE																								
SPSTAT																								
STATBD																								
THTRPLN																								
THTRBD																								
TRACK																								
TYPES																								
CHOLD																								

NOT COMMON BLOCKS

	CLOSCOR	CNACTIK	CODE01	CODE03	CODE05	CODE1B	COMMAND	COMMO	CONTROL	CORBOUN	COVAPLY	CRCDIES	CRCVNT	CRCRIL	CRCROSS	CRCSET	CRCTHNK	CRCTRAK	CRCZINT	CREATE	CREFIMI	DECDMS	DECRALO	DEIADD
APRK																								
ACFRAG																								
AFM																								
AGPD																								
AGPK																								
CLOK																								
COMOUT																								
COM PTR																								
COM SCS																								
CRCOM																								
CSTK																								
DATA																								
DEBUG																								
DLAGS																								
XTRACE																								
STEXT																								
FS																								
HALT																								
INITPR																								
IODEV																								
JUOPT																								
LIMITS																								
MASK																								
MODVAR																								
MXMIS																								
PATH																								
SAMPK																								
SAMPRS																								
SEMINFO																								
SPACE																								
SPSTAT																								
STATBD																								
THTRPLN																								
TRACK																								
TYPES																								
CHOLD																								

NO COMMON BLOCKS

DESTROY	DETECT	DGTSHX	DIGOUT	DLYACT	DMSDEC	DOCFILE	DGGTHNK	DROPRK	DROPP05	DROPP52	ENGAGE	ENTRYP	ENTSTAT	EXITP	FLDEL	FETCH	FLERUP	FLIFR	FINDBLK	FINDFIT	FINDII	FIRECHK	FETE
APK																							
ACFRAG																							
AFM																							
AGPD																							
AGPK																							
CLOK																							
COMOUT																							
COM PTR																							
COM SCS																							
CRCOM																							
CSTK																							
DATA																							
DEBUG																							
DFLAG																							
XTRACE																							
STEXT																							
FS																							
HALT																							
INTPTR																							
IODEV																							
JOPT																							
LIMITS																							
MASK																							
MODVAR																							
MXMIS																							
PATH																							
SAMPK																							
SAMPTR																							
SEMINFO																							
SPACE																							
SPSTAT																							
STATBD																							
THTRPLN																							
TRACK																							
TYPES																							
CHOLD																							

NO COMMON BLOCKS

Function	APK	ACFRAG	AFM	AGPD	AGPK	CLOCK	COMOUT	COM PTR	COM SCS	CRCOM	CSTK	DATA	DEBUG	DFLAGS	XTRACE	STEXT	FS	HALT	INITPR	IODEV	JOPT	LIMITS	MASK	MODVAR	MXMIS	PATH	SAMPK	SAMPTRS	SEMINFO	SPACE	SPSTAT	STATBD	THRPLN	TRACK	TYPES	CHOLD							
FLTGEOM																																											
FLWTYPE																																											
FLY																																											
FLYSEE																																											
FORMTGT																																											
FSDUMP																																											
FSINIT																																											
FURCHK																																											
GFTRK																																											
GFPTHS																																											
GIMME																																											
GNDLOOK																																											
GOGTEM																																											
GOTOAB																																											
HANDZPT																																											
HE XADD																																											
HE XCHZ																																											
HE XINV																																											
HE XMIT																																											
HE XMOVE																																											
HE XMULT																																											
HISTORY																																											
ICLPT																																											
HOLD																																											

NO COMMON BLOCKS

	HXDGTS	HXMITZ	IJ2HX	INDEX	INIT	INITACO	INRANGE	INSECT	INSERT	INTASIN	INTEND	INTFLY	INTZRC	IP II	IGH SUII	JLJ	JUGGLE	KUTHT	KOMPARE	KMLOC	LDIX	LNPLOT	LOADPI	LDSRADR
APPK																								
ACFRAG																								
AFM																								
AGPD																								
AGPK																								
CLOCK																								
COMOUT																								
COM PTR																								
COM SCS																								
CRCOM																								
DEBUG																								
DFLAGS																								
XTRACE																								
STEXT																								
FS																								
HALT																								
INITPTR																								
IODEV																								
JUOPT																								
LIMITS																								
MASK																								
MOVVAR																								
MXMIS																								
PATH																								
SAMPK																								
SAMPTRS																								
SEMINFO																								
SPACE																								
SPSTAT																								
STATBD																								
THTRPLN																								
TRACK																								
TYPES																								
CHOLD																								

NOT COMMON BLOCKS

Function	LTRF	MASKER	MESSAGE	MESBILD	NAYBOR	NEWMOVE	NEWPERC	NOWUCHI	NUKBLIND	OPTPTH	OTHRDAT	OUTA	OUTPRS	PACK	PAGE	PATCHC	PDOUT	PELARD	PERCEPT	PLAN	PLANOUT	PONDER	PREPAPU	PRIORITY
APK																								
ACFRAG																								
AFM																								
AGPD																								
AGPK																								
CLOCK																								
COMOUT																								
COMPTR																								
COMSCS																								
CRCOM																								
CSTK																								
DATA																								
DEBUG																								
DLRGS																								
XTRAGE																								
STEXT																								
FS																								
HALT																								
INPTR																								
IODEV																								
JOPT																								
LIMITS																								
MASK																								
MODVAR																								
MXMIS																								
PATH																								
SAMPK																								
SAMPTRS																								
SEMINFO																								
SPACE																								
SPSTAT																								
STATBD																								
THTRPLN																								
TRACK																								
TYPES																								
CHOLD																								

NO COMMON BLOCKS

Feature	PIPONER	PIRAND	PIRFE	READIL	RECCON	RECER	RECORD	REDEBRI	RELEASE	RELIST	RELOAD	RELOCAT	RESISE	RENDEVO	RESUPPLY	REVISE	RIFET	RIFEP	RIFER	RIABDB	RICORD	RIUMAKI	RIIRAID	RIUTAK
AAPK																								
ACFRAG																								
AFM																								
AGPD																								
AGPK																								
CLOCK																								
CUMOUT																								
COM PTR																								
COM SCS																								
CRCOM																								
CSTK																								
DATA																								
DEBUG																								
DLGAS																								
XTRACE																								
STEXT																								
FS																								
HALT																								
INTFR																								
IODEV																								
JOPT																								
LIMITS																								
MASK																								
MODVAR																								
MXMIS																								
PATH																								
SAMPK																								
SAMPTS																								
SEMINFO																								
SPACE																								
SPSTAT																								
STATBD																								
THTRPN																								
TRACK																								
TYPE																								
CHOLD																								

NOT COMMON BLOCKS

	AAPK	ACFRAG	AFM	AGPD	AGPK	CLOCK	COMOUT	COM PTR	COM SCS	CRCOM	CSK	DATA	DEBUG	DLG	XTRACE	STEXT	FS	HALT	INITR	JODEV	JOPT	LIMITS	MASK	MODVAR	MXMIS	PATH	SAMPK	SAMPTR	SEMINFO	SPACE	SPSTAT	STATBD	THTRPLN	TRACK	TYPE	CHOLD			
RITGTY																																							
RLWAVE																																							
RONDSEE																																							
ROUTER																																							
SAMATON																																							
SAMPHCM																																							
SAMSEF																																							
SAMWYPE																																							
SCHEDUL																																							
SCHTAB																																							
SDIGEST																																							
SEKENG																																							
SEKPP																																							
SEKKTAC																																							
SEKTFU																																							
SELECT																																							
SEMANT																																							
SETASSN																																							
SHRILL																																							
SHUFFLE																																							
SKSBTRK																																							
SNAP																																							
SRCHP																																							
SSL																																							

NO COMMON BLOCKS

STATPAK	STATBD	STAT	STATBD	THTRPLN	TRACK	TRACES	CHOLD
STATPAK							
STICK							
TABOUT							
TABOUTZ							
TERMACO							
TELYCRC							
TEGIGONE							
TEGIBEX							
TEGIBUS							
TEH2PS							
TEHTRPLN							
TEH2XY							
TEH2HX							
TEH2HX							
TEH2XY							
TEADH							
TEWER							
TEACE							
TEKCHEK							
TEYSHOT							
TEHME							
TEY2HX							
TEY2HX							
TEY2U							
APPK							
ACFRAG							
AFM							
AGPD							
AGPK							
CLOCK							
COMOUT							
COM PTR							
COM SCS							
CRCCOM							
CSTK							
DATA							
DEBUG							
DFLAGS							
XTRACE							
STEXT							
FS							
HALT							
INTRTR							
IODEV							
JOPT							
LIMITS							
MASK							
MODVAR							
MXMS							
PATH							
SAMPK							
SAMPTR							
SEMINFO							
SPACE							
SPSTAT							
STATBD							
THTRPLN							
TRACK							
TRACES							
CHOLD							

ALL COMMON BLOCKS

	UI SUB2	UMPHE	UNLINK	UNSNAP	UNSTAT	UOR SUB	UOLLOAD	WIPE OUT	WITHDRAW	XPAA	XPD	XPK	XSHIFT	XY2HX	YANK	YTRMBG	UNPAK	DBG READ	ICHECK	ITRAP	
AAPK																					
ACFRAG																					
AFM																					
AGPD																					
AGPK																					
CLOCK																					
COMOUT																					
COM PTR																					
COM SCS																					
CRCOM																					
CSTK																					
DATA																					
DEBUG																					
DLAGS																					
XTRACE																					
STEXT																					
FS																					
HALL																					
INIPTR																					
IODEV																					
JOPT																					
LIMITS																					
MASK																					
MQVAR																					
MXMIS																					
PATH																					
SAMPK																					
SAMPTS																					
SEMINFO																					
SPACE																					
SPSTAT																					
STATBD																					
THTRPN																					
TRACK																					
TYPES																					
CHOLD																					

NOT COMMON BLOCKS

Structure	MADEM	ARQUEUE	ARSEE	ABVSCOR	ABZPRC	ACCEPT	ACFRAG	ADASASS	ADDBLOK	AIRTHNK	ALLOBAT	ALLOFU	ALLOPAT	AMMUCHK	ASSIGN	ATKASES	ATTACK	AUTOPRI	AVAILBI	AZILIM	BADMOVE	BATCEAS	BATTCDV	BATTOUT
ARINFO	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ABQUEDB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ABSTATUS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ABVCR	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ACDB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ACDBUF	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ACODEVICE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ACRFLUST	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ACRAFTONAB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ACTAB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ADDUNK	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ADSIENDB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ALLOCATE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ADDR	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ARCFTSAW	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ARCFTSTATUS	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ATTACKBLOK	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
BOCSTAT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
BTRAYDIL	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
BTRAYSTAT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
BUFFER	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
C2	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
C2 SIDE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
COMMAND	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
CONSTBLOK	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
CORRIDOR BLOK	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
CRCSEE BLUE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
CRC SEERED	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
CRCSEES	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
CRCSEEBORD	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DADE	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DATBLOK	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DAT BUF	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DBCLASS BLOK	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
DIB	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

NO DATA STRUCTURES

ABINFO	BYALCOV	BYCNDPR	BYCONHO	BYCONLS	BYCONTC	BYENDPS	BYHEDUP	BYNOTRD	BYNWTRK	BYPASUP	BYPONER	BYPONFD	BYPONRL	BYPONRS	BYPONTM	BYTKCHK	BYUPDAT	CANCAIO	CANDTGT	CENTER	CFLYCRC	CHKCOV	CHKLAST	CLIST
ABQUEDB																								
ABSTASUS																								
ABVCR																								
ACDB																								
ACDBUF																								
ACDEVICE																								
ACRFTLIST																								
ACRFTONAB																								
ACTAB																								
ADLINK																								
ADSTEDB																								
ALOCATE																								
ADDR																								
ARCTSAW																								
ARCTSTATUS																								
ATTACK BLOK																								
BOCSTAT																								
BTRYDIL																								
BTRYSTAT																								
BUFFER																								
C2																								
C2 SIDE																								
COMMAND																								
CONSTBLOK																								
CORRIDOR BLOK																								
CRCSEERED																								
CRCSEERED BUE																								
CRCSEES																								
CRCSUBORD																								
DAQE																								
DATABLOK																								
DAT BUE																								
DBCLASS BLOK																								
DIB																								

NO DATA STRUCTURES

	CLOSCOR	CNACTTK	CODE01	CODE03	CODE05	CODE18	COMMAND	COMMO	CONTROL	CORBOUN	COVAPLY	CRCDIES	CRCEVNT	CRCKIL	CRCLOSS	CRCSEE	CRC7HINK	CRCTRAK	CRC2INT	CREATE	CRFLIML	DECDMS	DECRALO	DELAOD
ABINFO																								
ABQUEDB																								
ABSTATUS																								
ABVCR																								
ACDB																								
ACQBUF																								
ACQDEVICE																								
ACRFLIST																								
ACRFTONAB																								
ACTAB																								
ADDLINK																								
ADSLINK																								
ALLOCATE																								
ADDB																								
ARGFTSAW																								
ARGFTSTATUS																								
ATTACK BLOK																								
BOCSTAT																								
BTRYDIL																								
BTRYSTAT																								
BUFFER																								
C2																								
C2 SIDE																								
COMMAND																								
CONSTBLOK																								
CORRIDOR BLOK																								
CRCSEE BLUE																								
CRC SEERED																								
CRCSEES																								
CRCSUBORD																								
DAQE																								
DATBLOK																								
DAT BUF																								
DBCLASS BLOK																								
DIB																								

*NO DATA STRUCTURES

ABINFO	ABQUEDB	ABSTATUS	ABVCR	ACDB	ACDBUF	ACDEVICE	ACRFLIST	ACRFTONAB	ACTAB	ADDLINK	ADDITEDB	ALLOCATE	ADDB	ARCFTSAW	ARCFTSTATUS	ATTACK BLOK	BOCSTAT	BTRDIL	BTRSTAT	BUFFER	C2 SIDE	COMMAND	CONSTBLOK	CORRIDOR BLOK	CRCSEE BLUE	CRCSEED	CRCSEES	CRCSUBORC	DAOE	DATABLOK	DAT BUF	DECLASS BLOK	DIB			
FLTGEOM																																				
FLTWYPE																																				
FLY																																				
FLYSEE																																				
FORMATGT																																				
FSDUMP																																				
FSINIT																																				
FUELCHK																																				
GETEX																																				
GETPRS																																				
GIMME																																				
GNDLOOK																																				
GOGETEM																																				
GOTOAB																																				
HANDZPT																																				
HEXADD																																				
HEXCHZ																																				
HEXINV																																				
HEXMLT																																				
HEXMOVE																																				
HEXMULT																																				
HISTORY																																				
HLPNT																																				
HOLD																																				

*NO DATA STRUCTURES

RTGTYP	RIWAVE	RONDSEE	ROUTER	SAMATON	SAMPRCM	SAMSEE	SAMWYPE	SCHEDUL	SCHTAB	SDIGEST	SEEKENG	SEEKP	SEEKTAC	SEEKTUFU	SELECT	SEMANT	SETASSN	SHRILL	SHUFFLE	SKSSTRK	SNAP	SRCHPL	SSIL
ABINFO																							
ABQUEDB																							
ABSTATUS																							
ABVCR																							
ACDB																							
ACQBUF																							
ACODEVICE																							
ACRFLUST																							
ACRFTONAB																							
ACTAB																							
ADLINK																							
ADSITEB																							
ALLOCATE																							
ADDB																							
ARCTSAW																							
ARCTSTATUS																							
ATTACK BLOK																							
BOCSTAT																							
BTRYDIL																							
BTRYSTAT																							
BUFFER																							
C2																							
C2 SIDE																							
COMMAND																							
CONSTBLOK																							
CORRIDOR BLOK																							
CRCSEE BLUE																							
CRCSEERED																							
CRCSEES																							
CRCSUBCRD																							
DAGE																							
DATBLOK																							
DAT BUF																							
DBCLASS BLOK																							
DIB																							

*NO DATA STRUCTURES

STATPAK	STATPAK
STICK	
TABOUT	
TABOUT2	
TERMACO	
TFLYCRC	
TGTGONE	
TGTHX	
TGTLIST	
THH2PS	
THTRPLN	
THX2XY*	
TH2HX*	
TLL2HX*	
TLL2XY*	
TOADIL	
TOWER	
TRACE*	
TRKCHK	
FRYSHOT	
TTIME*	
TXY2HX*	
TXY2HXL*	
TXY2LL*	
ABINFO	
ARQUEDB	
ABSTATUS	
ABVCR	
ACDB	
ACDBUF	
ACDEVICE	
ACRFLUST	
ACFTONAB	
ACTAB	
ADLINK	
ADSIEDB	
ALLOCATE	
ADDB	
ARCFTSAW	
ARCFTSTATUS	
ATTACK BLOK	
BRYDIL	
BRYSTAT	
BUFFER	
C2 SIDE	
COMMAND	
CONSTBLOK	
CORRIDOR BLOK	
CRCSEEBLUE	
CRCSEERED	
CRCSEES	
CRCSUBORD	
DAOE	
DATBLOK	
DAT BUF	
DBCLASS BLOK	
DIB	

*NO DATA STRUCTURES

DIL	DUMMY BLOCK
ENGAGE	EVENT
EVENT	EYEBALL
EYEBALL	FAKTGBLOK
FAKTGBLOK	FIRE UNIT
FIRE UNIT	FLAKTBLOK
FLAKTBLOK	FMFLT DB
FMFLT DB	FOREST
FOREST	FORMATION BLOK
FORMATION BLOK	FORMATION BLOK
FORMATION BLOK	FORGT BUFFER
FORGT BUFFER	HEAD BLOK
HEAD BLOK	HEX
HEX	HEXBLOK
HEXBLOK	HEXLEU
HEXLEU	HEXLINK
HEXLINK	LEFTREE
LEFTREE	LINK
LINK	LOAD
LOAD	MESSAGE
MESSAGE	MUN
MUN	NOAVAIL BLOK
NOAVAIL BLOK	ORDERS
ORDERS	PAL
PAL	PAYBUF
PAYBUF	PAYLDBLOK
PAYLDBLOK	PAYLOAD
PAYLOAD	PER LIST
PER LIST	PLAYER BUFFER
PLAYER BUFFER	PL BUFFER
PL BUFFER	PLYST

BDALT
BDLEX
BDLRK
BDPARS
BLKDAT
BNCMDPPH
BNCUNHD
BNCUNLS
BNCUNTC
BNALLE
BNNOTHD
BNNWTRK
BNPONBB
BNPONBD
BNPONDA
BNPONEP
BNPOMFA
BNPONFD
BNPONSS
BNRECOV
BNCUNTK
BTNASIN
BINZCRC
BTRYINK

	DIL	DUMMY BLOCK	ENGAGE	EVENT	EYEBALL	FAKTGBLOK	FDBDBLOK	FIRE UNIT	FLAKTBLOK	FLTDB	FMFLT DB	FOREST	FORMATION BLOK	FORMATION BLOK	FORMATION BLOK	HEAD BUFFER	HEX	HEXBLOK	HEXELV	HEXLINK	LEFTREE	LINK	LOAD	MESSAGE	MUN	NOAVAIL BLOK	ORDERS	PAL	PAYBUB	PAYLDBLOK	PAYLOAD	PER LIST	PLAYER BUFFER	PL BUFFER	PLYLIST				
BYALCOV																																							
BYCMDPR																																							
BYCONHD																																							
BYCONLS																																							
BYCONIC																																							
BYENDPS																																							
BYHE DUP																																							
BYNOTRO																																							
BYNWTRK																																							
BYPASUP																																							
BYPNER																																							
BYPONRD																																							
BYPONRI																																							
BYPONRS																																							
BYPONTM																																							
BYTKCHK																																							
BYUPDAT																																							
CANCAIO																																							
CANDTGT																																							
CENTER																																							
CFLYCRC																																							
CHRCOV																																							
CHKLAST																																							
CLUST																																							

	CLOSCOR	CNACTTK	CODE01	CODE03	CODE05	CODE18	COMMAND	COMMO	CONTROL	COHBOUN	COVAPLY	CRCDH S	CRCVNT	CRCRIL	CRCLOSS	CRCSEE	CRCTHNK	CRCTRAK	CRCZINT	CREATE	CRETML	DECDSMS	DECRALO	DEIADD
DIL																								
DUMMY BLOCK																								
ENGAGE																								
EYEBALL																								
EVENT																								
FAKTGLOB																								
FDBDBLOK																								
FIRE UNIT																								
FLTAKTBLOK																								
FLTDB																								
FMTL DB																								
FOREST																								
FORMATION																								
FORMATION BLOK																								
FORGT BUFFER																								
HEAD BLOK																								
HEXBLOK																								
HEXLEV																								
HEXLINK																								
LEFTREE																								
LINK																								
LOAD																								
MESSAGE																								
MUN																								
NOAVAIL BLOK																								
ORDERS																								
PAL																								
PAYBUF																								
PAYLDBLOK																								
PAYLOAD																								
PER LIST																								
FLAYER BUFFER																								
PL BUFFER																								
PLYST																								

DIL	DUMMY BLOCK	ENGAGE	EVENT	EYEBALL	FAKTGBLOK	FDRJBLOK	FIRE UNIT	FLTAKTBLOK	FLTDB	FMELT DB	FOREST	FORMATION BLOK	FORMATION BLOK	FORTGT BUFFER	HEAD BLOCK	HEX	HEXBLOK	HEXLEV	HEXLINK	LE-TREE	LINK	LOAD	MESSAGE	MUN	NOAVAIL BLOK	ORDERS	PAL	PAYBUF	PAYLOADBLOK	PAYLOAD	PER LIST	PLAYER BUFFER	PL BUFFER	PLYST					
DESTROY																																							
DETECT																																							
DGTSHX																																							
DII OUT																																							
DIYACT																																							
DMSDLC																																							
DOGFITE																																							
DOGTHNK																																							
DROPBK																																							
DROPPS																																							
DROPPSZ																																							
ENGAGE																																							
ENTRYP																																							
ENSTAT																																							
EXITP																																							
FELDEL																																							
FETCH																																							
FILERUP																																							
FILTER																																							
FINDBLK																																							
FINDFLT																																							
FINDIT																																							
FIRECHK																																							
FLUTE																																							

	FLTGEOM	FLTWYPE	FLY	FLYSEE	FORMTGT	FSDUMP	FSINIT	FUELCHK	GETHEX	GETPIRS	GIMME	GNDLOOK	GOGITEM	GOTOAB	HANDZPT	HEXADD	HEXCHZ	HEXINV	HEXMLT	HEXMOVE	HEXMULT	HISTORY	HTPNT	HOLD
DUMMY BLOCK																								
ENGAGE																								
EVENT																								
EYEBALL																								
FAKTGBLOK																								
FDBDBLOK																								
FIRE UNIT																								
FLAKTBLOK																								
FLTDB																								
FMLT DB																								
FOREST																								
FORMATION																								
FORMATION BLOK																								
FORGT BUFFER																								
HEAD BLOK																								
HEX																								
HEXBLOK																								
HEXLEV																								
HEXLINK																								
LEFTREE																								
LINK																								
LOAD																								
MESSAGE																								
MUN																								
NOAVAIL BLOK																								
ORDERS																								
PAL																								
PAYBUF																								
PAYDDBLOK																								
PAYLOAD																								
PER LIST																								
PLAYER BUFFER																								
PL BUFFER																								
PLYST																								

	LFREE	MASKER	MESSAGE	MESBILD	NAYBOR	NEWMOVE	NEWPERC	NOWUICIT	NUKBUND	OPTPTH	OTHRDAT	OUTA	OUTPERS	PACK	PAGE	PATDEC	POLOUT	PELADD	PEWCEPT	PLAN	PLANOUT	PONDER	PREPAFU	PRIORITY
DIL																								
DUMMY BLOCK																								
ENGAGE																								
EVENT																								
EYEBALL																								
FAKTGBLOK																								
FDBDBLOK																								
FIRE UNIT																								
FLAKTBLOK																								
FLTOB																								
FMFLT DB																								
FOREST																								
FORMATION BLOK																								
FORMATION BLOK																								
FORGT BUFFER																								
HEAD BLOK																								
HEX BLOK																								
HEX																								
HEXBLOK																								
HEXLEV																								
HEXLINK																								
LEFTREE																								
LINK																								
LOAD																								
MESSAGE																								
MUN																								
NOAVAIL BLOK																								
ORDERS																								
PAL																								
PAYBUF																								
PAYDDBLOK																								
PAYLOAD																								
PER LIST																								
PLAYER BUFFER																								
PL BUFFER																								
PLYST																								

	DIL	DUMMY BLOCK	ENGAGE	EVENT	EYBALL	FAKTGBLOK	FDBDBLOK	FIRE UNIT	FLTAKTBLOK	FLTOB	FMELT DB	FOREST	FORMATION	FORMATION BLOK	FORMATION BLOK	FORGT BUFFER	HEAD BLOCK	HEX	HEXBLOK	HEXLEV	HEXLINK	LEFTREE	LINK	LOAD	MESSAGE	MUN	NOAVAIL BLOK	ORDERS	PAL	PAYBUF	PAYLDBLOK	PAYLOAD	PER LIST	PLAYER BUFFER	PLYST			
PIPONER																																						
PTRAND																																						
PTREE																																						
READR																																						
RECCON																																						
REER																																						
RECORD																																						
REDEBRF																																						
RELEASE																																						
RELIST																																						
RELOAD																																						
RELOCAT																																						
RELSIT																																						
RENDEVD																																						
RESUPPLY																																						
REVISE																																						
RITET																																						
RITEP																																						
RITER																																						
RLABDB																																						
RUCORD																																						
RUFMAKT																																						
RIBRAID																																						
RITGTAK																																						

	RUTGTYP	RIWAVI	RONDSEI	ROOTER	SAMATON	SAMPHEM	SAMSEI	SAMWVPI	SCHRDH	SCHTAB	SIDGSI	SEKENG	SEIKP	SEKAC	SEKED	SEKCT	SEMANT	SEJASSN	SHRILL	SHUJLI	SK-SIBIK	SNAP	SRCHPT	SSLI
DIL																								
DUMMY BLOCK																								
ENGAGE																								
EVENT																								
EYBALL																								
FAKTGBLOK																								
FDBGBLOK																								
FIRE UNIT																								
FLAKTBLOK																								
FLTD8																								
FMILT DB																								
FOREST																								
FORMATION																								
FORMATION BLOK																								
FORTGT BUFFER																								
HEAD BLOCK																								
HEX																								
HEXBLOK																								
HEXLEV																								
HEXLINK																								
LEFTREE																								
LINK																								
LOAD																								
MESSAGE																								
MUN																								
NOAVAIL BLOK																								
ORDERS																								
PAL																								
PAYBUF																								
PAYLDBLOK																								
PAYLOAD																								
PER LIST																								
PLAYER BUFFER																								
PL BUFFER																								
PLYST																								

STATPAK	STICK	TABOUT	TABOUT2	TERMACO	TFLYCRC	TGTGONE	TGTHEX	TGTUJST	THH2PS	THTRPIN	THX2XY	TH2HX	TH2HX	TH2HX	TH2HX	TADIL	TOWER	TRACE	TRKCHK	TRYSHOT	TIME	TXZHX	TXZHX	TXZHX	TXZHX
DIL																									
DUMMY BLOCK																									
ENGAGE																									
EVENT																									
EYEBALL																									
FAKTGBLCK																									
FDBDBLCK																									
FIRE UNIT																									
FLTAKTBLOK																									
FLTDB																									
FMFLT DB																									
FOREST																									
FORMATION																									
FORMATION BLOK																									
FORTGT BUFFER																									
HEAD BLOK																									
HEX																									
HEXBLOK																									
HEXELEV																									
HEXLINK																									
LEFTREE																									
LINK																									
LOAD																									
MESSAGE																									
MUN																									
NOAVAIL BLOK																									
ORDERS																									
PAL																									
PAYBUF																									
PAYLDBLOK																									
PAYLOAD																									
PER LIST																									
PLAYER BUFFER																									
PL BUFFER																									
PLYST																									

	PROFIBD BLOK	PU BUFFER	QUESTAT	QUESTS	RAID BLOK	READY QUE	SB	SDB	SEE BUF	SEER	SOURCE	STDBLOK	SUB	SUBLIST	SUBTYPE	TARGET BLOK	TARGET BUFFER	TGT LIST BLOK	TGT PTR	TTD BLOK	WAVE BLOK	WINGMAN	
MADM																							
ABQUEUE																							
ABSEE																							
ABVSCOR																							
AB2CRC																							
ACCEPT																							
ACFRAG																							
ADASASS																							
ADDBLOK																							
AIRTHNK																							
ALLOBAT																							
ALLOFU																							
ALLOPAT																							
AMMOCHK																							
ASSIGN																							
ATKASES																							
ATTACK																							
AUTOPRI																							
AVAILBI																							
AZILIM																							
BADMOVE																							
BATCEAS																							
BATTCOV																							
BATTOUT																							

	PROFLED BLOK	PU BUFFER	QUESTAT	QUEST	RAID BLOK	READY QUE	SB	SDB	SEE BUF	SEER	SOURCE	STDBLOK	SUB	SUBJUST	SUBTYPE	TARGET BLOK	TARGET BUFFER	TARGET LIST BLOK	TGT PTR	TTD BLOK	WAVE BLOK	WINGMAN	
CLOSCOR																							
CNACTTK																							
CODE01																							
CODE03																							
CODE05																							
CODE18																							
COMMAND																							
COMMO																							
CONTROL																							
CORBOUN																							
COVAPLY																							
CRCDIES																							
CRCEVNT																							
CRCKIL																							
CRCLOSS																							
CRCSEE																							
CRCTHINK																							
CRCTRAK																							
CRC2INT																							
CREATE																							
CRFLTML																							
DECDMS																							
DECRALO																							
DELADD																							

DESTROY	DETECT	DGTSHX	DILOUT	DIYACT	DMSDEC	DOGFITE	DOGTHNK	DROPBK	DROPP0S	DROPPS2	ENGAGE	ENTRYP	ENSTAT	EXITP	FELDEL	FETCH	FILERUP	FILTER	FINDBLK	FINDLIT	FINDIT	FIRECHK	FITE
PROFIED BLOK																							
PU BUFFER																							
QUESTAT																							
QUES																							
RAID BLOK																							
READY QUE																							
SB																							
SDB																							
SEE BUF																							
SEER																							
SOURCE																							
STDBLOK																							
SUB																							
SUBLIST																							
SUBTYPE																							
TARGET BLOK																							
TARGET BUFFER																							
TARGET LIST BLOK																							
TGT PTRAE																							
TTD BLOK																							
WAVE BLOK																							
WINGMAN																							

PROFILD BLOK	PU BUFFER	QUESTAT	QUEST	RAID BLOK	READY QUE	S8	SDB	SEE BUF	SEER	SOURCE	STDBLOK	SUB	SUBLIST	SUBTYPE	TARGET BLOK	TARGET BUFFER	TARGET LIST BLOK	TGT PTREE	TTO BLOK	WAVE BLOK	WINGMAN	
HXDGTS																						
HXM1 T2																						
IJ2HX																						
INDIX																						
INIT																						
INITACU																						
INRANGE																						
INSFCT																						
INSERT																						
INTASIN																						
INTFIND																						
INTRFLY																						
INTZCRC																						
IPR																						
JGESUIT																						
JTJ																						
JUGGLE																						
KIFLIT																						
KOMPARE																						
LCMILOC																						
LINEX																						
LNPILOT																						
LOADPI																						
LOSRAOR																						

LIBRE	MASKER	MESSAGE	MESBID	NAYBOR	NEWMOVE	NEWPERC	NOOWICIT	NUKBIND	OPTPH	OTHRDAT	OUTA	OUTPERS	PACK	PAGE	PATDEC	POCOUT	PLADD	PERCPT	PLAN	PLANOUT	PONDER	PREPARE	PRIORITY
PROFIBD BLOK																							
PU BUFFER																							
QUESTAT																							
QUEST																							
RAID BLOK																							
READY QUE																							
SB																							
SDB																							
SEE BUE																							
SEER																							
SOURCE																							
STDBLOK																							
SUB																							
SUBST																							
SUBTYPE																							
TARGET BLOK																							
TARGET BUFFER																							
TARGET LIST BLOK																							
TGT PTREE																							
TTD BLOK																							
WAVE BLOK																							
WINGMAN																							

RUIGTYP	RIWAVE	RINDSEE	ROUTER	SAMATON	SAMPROM	SAMSET	SAMWYPT	SCHDUL	SCHTAB	SDIGEST	SEKENG	SEKRP	SEKTAC	SEKTFU	SEKTT	SEMANT	SETASSN	SHRKH	SHUFLU	SKSBRK	SNAP	SCHUPI	SSU
PROFLED BLOK																							
PU BUFFER																							
QUESTAT																							
QUEST																							
RAID BLOK																							
READY QUE																							
SB																							
SDB																							
SEE BUF																							
SEER																							
SOURCE																							
STDBLOK																							
SUB																							
SUBJST																							
SUBTYPE																							
TARGET BLOK																							
TARGET BUFFER																							
TARGET LIST BLOK																							
TGT PTREE																							
TTD BLOK																							
WAVE BLOK																							
WINGMAN																							

STAIPAK	SJCK	TABOUT	TABOUTZ	TFRMACO	TFYGRIC	TGAGONI	TGDFX	TGJNST	TIR2PS	TIRPPIN	TIX2XY	TIZHX	TIZHX	TIZXY	TOADH	TOWER	TWACT	TWRCHK	TRYSHOT	TUMI	TYZHX	TYZHX	TYZII
PROFIED BLOK																							
PU BUFFER																							
QUESTAT																							
QUEST																							
RAID BLOK																							
READY QUE																							
SB																							
SDB																							
SEE BUF																							
SEER																							
SOURCE																							
STDBLOK																							
SUB																							
SUBLIST																							
SUBTYPE																							
TARGET BLOK																							
TARGET BUFFER																							
TARGET LIST BLOK																							
TGT PTRFE																							
TJD BLOK																							
WAVE BLOK																							
WINGMAN																							

UI SUB7	UMPIRE	UNLINK	UNSNAP	UNSTAT	UOISUB	UOULOAD	USPTOUT	WITHDRAW	XPAA	XPD	XPK	XSHFT	XYZHX	YANK	UTMRG	UNPACK	DBGREAD	ICHECK
PROFLED BLOK																		
PU BUFFER																		
QUESTAT																		
QUESTS																		
RAID BLOK																		
READY QUE																		
SB																		
SDB																		
SEE BUF																		
SEER																		
SOURCE																		
STDBLOK																		
SUB																		
SUBLIST																		
SUBTYPE																		
TARGET BLOK																		
TARGET BUFFER																		
TARGET LIST BLOK																		
TGT PFREE																		
TTD BLOK																		
WAVE BLOK																		
WINGMAN																		

DISTRIBUTION LIST

DEPARTMENT OF DEFENSE

Armed Forces Staff College
ATTN: Reference & Technical Svcs Br

Assistant Secretary of Defense
International Security Affairs
ATTN: European & NATO Affairs
ATTN: Policy Plans & NSC Affairs

Assistant Secretary of Defense
Comm, Cmd, Cont & Intell
ATTN: Surveillance & Warning Sys

Assistant to the Secretary of Defense
Atomic Energy
ATTN: Executive Assistant

Commander-in-Chief, Atlantic
ATTN: N-22

Commander-in-Chief, Pacific
ATTN: J-634
ATTN: J-54
ATTN: J-32

Defense Intelligence Agency
ATTN: DIA/VPA-2, Fed Res Div
ATTN: DB-6
5 cy ATTN: DB-4

Defense Nuclear Agency
ATTN: STNA
ATTN: NATD
ATTN: NATA
4 cy ATTN: TITL

Defense Technical Information Center
12 cy ATTN: DD

Field Command
Defense Nuclear Agency
ATTN: FCPR

Field Command
Defense Nuclear Agency
Livermore Branch
ATTN: FCPR

Intelligence Center, Pacific
ATTN: COMIPAC

Interservice Nuclear Weapons School
ATTN: TTV

Joint Chiefs of Staff
ATTN: J-3
ATTN: J-5, Strategy Division, W. McClain
ATTN: J-5, Nuclear Division
ATTN: J-3, Strategic Operations Div
ATTN: SAGA/SSD
ATTN: SAGA/SFD

National Defense University
ATTN: NWCLB-CR
ATTN: ICAF, Tech Lib

DEPARTMENT OF DEFENSE (Continued)

Joint Strat Ig^t Planning Staff
ATTN: JL
ATTN: JLA
ATTN: JP
2 cy ATTN: JLTW-2

Director
NET Assessment
Office of the Secretary of Defense
ATTN: Military Assistants

U.S. European Command
ATTN: J-2
ATTN: J-3
ATTN: J-LW
ATTN: J-5NPG
ATTN: J-6
ATTN: J-2-ITD

U.S. National Military Representative
SHAPE
ATTN: U.S. Documents Officer

Under Secretary of Defense for Rsch & Engrg
ATTN: Strategic & Space Sys (OS)
ATTN: Tactical Warfare Programs

DEPARTMENT OF THE ARMY

Deputy Chief of Staff for Ops & Plans
Department of the Army
ATTN: DAMO-ZXA
ATTN: DAMO-NC

Harry Diamond Laboratories
Department of the Army
ATTN: DELHD-N-P

U.S. Army Comd & General Staff College
ATTN: ACQ, Library Div

Army Concepts Analysis Agency
ATTN: C:SA-ADL

Commander-in-Chief
U.S. Army Europe and Seventh Army
ATTN: AEAGC
ATTN: AEAGE

U.S. Army Nuclear & Chemical Agency
ATTN: Library

U.S. Army TRADOC Sys Analysis Actvy
ATTN: ATAA-TAC

U.S. Army War College
ATTN: Library

U.S. Military Academy
Department of the Army
ATTN: Document Library

DEPARTMENT OF THE NAVY

Naval Postgraduate School
ATTN: Code 1424, Library

Naval Research Laboratory
ATTN: Code 1240

Naval Sea Systems Command
ATTN: SEA-09G53
ATTN: SEA-643

Naval Surface Weapons Center
ATTN: Code F31

Naval War College
ATTN: Document Control

Nuclear Weapons Tng Group, Pacific, Dept of Navy
ATTN: Document Control

Nuclear Weapons Tng Group, Atlantic, Dept of Navy
ATTN: Document Control

Office of Naval Research
ATTN: Technical Info Services

DEPARTMENT OF THE AIR FORCE

Air Force Systems Command
ATTN: DL
ATTN: XR
ATTN: SD

Air Force Weapons Laboratory
Air Force Systems Command
ATTN: SUL

Air University Library
Department of the Air Force
ATTN: AUL-LSE

Assistant Chief of Staff
Intelligence
Department of the Air Force
ATTN: INA

Deputy Chief of Staff
Research, Development, & Acq
Department of the Air Force
ATTN: AFRDQI

Foreign Technology Division
Air Force Systems Command
ATTN: CCN
ATTN: TQTM
ATTN: SDN

Strategic Air Command
Department of the Air Force
ATTN: NR
ATTN: ADWN
ATTN: XP
ATTN: NRI, STINFO Library
ATTN: DO

U.S. Air Force, Pacific
ATTN: XP

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: USAFE/DEX

DEPARTMENT OF THE AIR FORCE (Continued)

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: USAFE/DOT

Commander-in-Chief
U.S. Air Force in Europe
ATTN: USAFE/INA

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: USAFE/INT

Commander-in-Chief
U.S. Air Forces in Europe
ATTN: USAFE/XPX

USAF Tactical Fighter Weapons Center
ATTN: Commander

DEPARTMENT OF ENERGY

Department of Energy
Albuquerque Operations Office
ATTN: CTID

OTHER GOVERNMENT AGENCY

Central Intelligence Agency
ATTN: OSWR/NED

DEPARTMENT OF ENERGY CONTRACTORS

Lawrence Livermore National Lab
ATTN: Tech Info Dept, Library

Los Alamos National Lab
ATTN: MS 364
ATTN: M/S634, T. Dowler

DEPARTMENT OF DEFENSE CONTRACTORS

Advanced International Studies Institute
ATTN: M. Harvey

Aerospace Corp
ATTN: Library

BDM Corp
ATTN: J. Braddock
4 cy ATTN: M. Filteau
4 cy ATTN: T. Hawkins
4 cy ATTN: L. Elfes
4 cy ATTN: B. Macaleer
4 cy ATTN: J. Phillips

JAYCOR
ATTN: R. Sullivan

Kaman Sciences Corp
ATTN: F. Shelton

Kaman Sciences Corp
ATTN: E. Daus

Kaman Tempo
ATTN: DASIAC

Pacific-Sierra Research Corp
ATTN: G. Lang
ATTN: H. Brode

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Pacific-Sierra Research Corp
ATTN: D. Gormley

R & D Associates
ATTN: R. Port
ATTN: P. Haas
2 cy ATTN: Document Control
2 cy ATTN: S. Borjon

R & D Associates
ATTN: J. Thompson

Science Applications, Inc
ATTN: Document Control
ATTN: J. Warner
ATTN: M. Drake

Science Applications, Inc
ATTN: W. Layson
ATTN: Document Control

DEPARTMENT OF DEFENSE CONTRACTORS (Continued)

Science Applications, Inc
ATTN: D. Kaul
ATTN: L. Goure

System Planning Corp
ATTN: J. Douglas

Tetra Tech, Inc
ATTN: F. Bothwell

TRW Defense & Space Sys Group
ATTN: N. Lipner

TRW Defense & Space Sys Group
ATTN P. Dai

ED
88