

AD-A084 053

SOC INTEGRATED SERVICES INC SAN DIEGO CA

F/G 9/2

STAMMER2 PRODUCTION SYSTEM FOR TACTICAL SITUATION ASSESSMENT. V--ETC(U)

OCT 79 D C MCCALL, P H MORRIS, D F KIBLER

N00123-76-C-0172

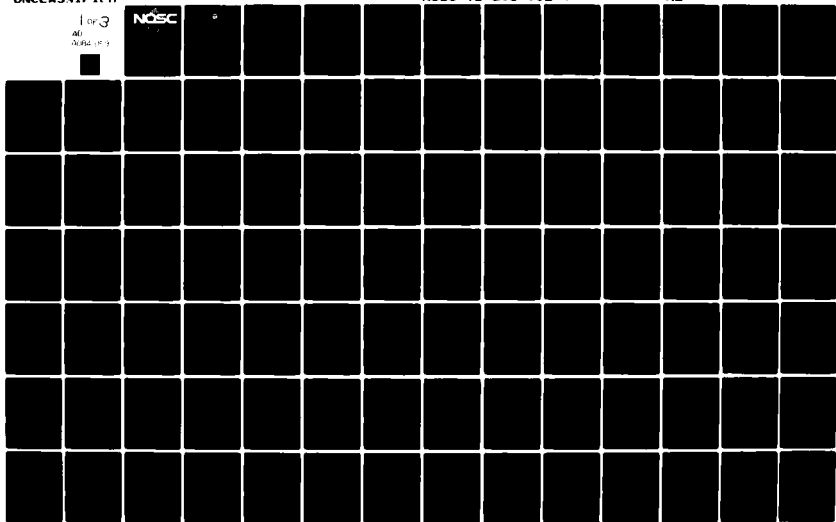
UNCLASSIFIED

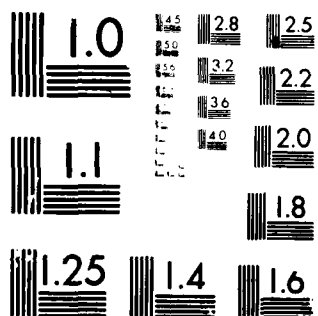
NOSC-TD-298-VOL-2

NL

1 of 3
AD
00000000

NOSC





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A 084053
NOSC TD 298

LEVEL ~~III~~

12

NOSC

DTIC
ELECTE
MAY 9 1980

A084038

NOSC TD 298

C

Technical Document 298

STAMMER2 PRODUCTION SYSTEM FOR TACTICAL SITUATION ASSESSMENT

Volume 2 — Code
(Volume 1 consists of the design description)

DC McCall (NOSC Task Leader)
PH Morris, DF Kibler, RJ Bechtel
(SDC Integrated Services)
Contract N00123-76-C-0172

October 1979

Prepared for
Naval Electronic Systems Command (NAVELEX 330)
Washington DC 20360

Approved for public release; distribution unlimited

NAVAL OCEAN SYSTEMS CENTER
SAN DIEGO, CALIFORNIA 92152

80 5 7 061

FILE COPY



NAVAL OCEAN SYSTEMS CENTER, SAN DIEGO, CA 92152

AN ACTIVITY OF THE NAVAL MATERIAL COMMAND

RR GAVAZZI, CAPT, USN

Commander

HL BLOOD

Technical Director

ADMINISTRATIVE INFORMATION

Work was performed by the Tactical Command and Control Division (Code 824) as a part of the Tactical Situation Assessment (TSA) problem under Program Element 62721N, Project F21201, Task Area XF21201100 (NOSC 824-CC18). This TSA task is a part of the Command Control Block Program sponsored by NAVELEX, Code 330—the Command and Control Division of Research and Technology Directorate, NAVELEX Code 03.

This document was written by PH Morris, DF Kibler, and RJ Bechtel, of SDC Integrated Services, under Contract N00123-76-C-0172. It covers work from June through September 1979 and was approved for publication 28 November 1979.

Released by
RC Kolb, Head
Tactical Command and
Control Division

Under authority of
JH Maynard, Head
Command Control—Electronic Warfare
Systems and Technology Department

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(18) NOSC

(19) TD-298-VOL-

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NOSC Technical Document 298 (TD 298)	2. GOVT ACCESSION NO. AD-A084 053	3. RECIPIENT'S CATALOG NUMBER (9)
4. TITLE (and Subtitle) (6) STAMMER2 PRODUCTION SYSTEM FOR TACTICAL SITUATION ASSESSMENT Volume 2a Code	5. TYPE OF REPORT & PERIOD COVERED Technical document June through September 1979	
7. AUTHOR(s) DC McCall (NOSC Task Leader) PH Morris, DF Kibler, RJ Bechtel (SDC Integrated Services)	8. CONTRACT OR GRANT NUMBER(s) (15) N00123-76-C-0172	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Systems Center San Diego CA 92152	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS (17) 62721N, F21201 XF21201100 (NOSC 824-CC18)	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command (NAVELEX 330) Washington DC 20360	12. REPORT DATE (11) October 1979	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (10) D.C. McCall P.H. Morris	13. NUMBER OF PAGES 214	
16. DISTRIBUTION STATEMENT (of this Report) D.F. Kibler R.J. Bechtel Approved for public release; distribution unlimited	15. SECURITY CLASS. (of this report) Unclassified	
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) STAMMER2 Production systems Rule-based inference systems Confidence factors Tactical analyses Correlation techniques Merchant detection		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) STAMMER2 is a revised version of STAMMER, a System for Tactical Assessment of Multisource Messages, Even Radar. STAMMER was created as part of an investigation of new correlation methodologies, and served as a testbed for explorations of applications of rule-based inference systems to the tactical situation assessment (TSA) problem. STAMMER concentrated on the specific task of merchant detection from radar and external messages. Experience with STAMMER revealed areas for improvement, which have led to the creation of STAMMER2. In addition to several changes in the underlying rule mechanisms used, the enhancements found in STAMMER2 arose out of a desire for greater generality and flexibility in the demonstration system, the explanation system, and the range of		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

395265

B

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (Continued)

acceptable inputs to the system. STAMMER2 should prove to be a more useful system for testing various rule/scenario collections. During the development that led to STAMMER2, further issues in the design of rule-based inference systems for use in support of C3 activities have become apparent and they are discussed. This volume consists of the code. Volume 1 consists of the design description.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

Table of Contents

Organization of this Volume	ii
Index to Functions	1
Program Listings	11
Function Calling Sequence	175
Function Descriptions	181

Accession For	
NTIS	☒
DDC TAB	☐
Unannounced	☐
Justification	☐
By _____	
Distribution/ _____	
Availability Codes	
Dist	Availand/or special
A	

Organization of this Volume

Dealing with the code is always a problem. While it is, in a sense, the fruit of long labor, no one likes to list it out and include it in the prose report which is almost inevitably the final output of a programming effort. However, there are always questions which can be answered only by appeal to the code itself, making its availability a necessity, and its inclusion in a final report desirable.

We have dealt with the problem by making the code a separate volume of this final report. Those who need or want to see the messy details are welcome, while others can ignore them without qualms, and without the need to carry around listings that will never be used.

Code by itself, even when commented, can be particularly uninformative. We have attempted to make use of software engineering tools provided by INTERLISP to make wading through the program itself somewhat easier. After the listing, we include a tree which represents the calling sequence of the functions that make up STAMMER. It is important to note that not all of the functions in the code will be included in this tree, as some are intended to be top-level calls only, rather than called by other functions. However, this does give an idea of the flow through the functions during normal execution.

Following the calling sequence tree, we provide a brief description of each function, in alphabetical order. This

description includes calling and called-by information, along with variable binding information.

The insert before the code proper is a cross-reference from an alphabetical listing of function names to the name of the file in which they are included. This should make looking up code much easier, since the files are organized functionally, rather than lexically. This index also includes all properties and variables which are set in the file, though these are not given reference numbers.

Summary for files: <RBECHTAL>CONFIDENCE..23 24-Jul-79 13:44:49 PAGE 1
 <PMORRIS>DSPLA.LSP.88 10-Aug-79 16:19:32
 <PMORRIS>FORK.LSP.19 18-Dec-78 16:30:43
 <RBECHTAL>HASHER..38 7-Aug-79 19:03:11
 <RBECHTAL>INTERP..37 28-Aug-79 21:06:16
 <RBECHTAL>MANIPULATE..20 6-Aug-79 17:32:02
 <RBECHTAL>MEMORY..17 6-Aug-79 19:06:57
 <RBECHTAL>MSGMTR..27 23-Aug-79 17:56:55
 <DKIBLER>NEWEXP.LSP.34 28-Aug-79 11:42:07
 <DKIBLER>ORACLE.LSP.40 8-Aug-79 09:11:09
 <DKIBLER>PLAT.LSP.48 6-Aug-79 11:01:49
 <PMORRIS>QH.LSP.72 21-Aug-79 12:09:01
 <RBECHTAL>RULES..29 27-Aug-79 21:39:39
 <PMORRIS>STREAM.LSP.37 6-Aug-79 20:15:20
 <RBECHTAL>TOPELVEL..13 21-Aug-79 11:08:03

<ATTIS>		NEWEXP	ifprop: QHPRODS
<EXPLTREE>		NEWEXP	ifprop: QHPRODS
<ID2>		NEWEXP	ifprop: QHPRODS
<IDAMP2>		NEWEXP	ifprop: QHPRODS
<IDAMPIS>		NEWEXP	ifprop: QHPRODS
<IDIS>		NEWEXP	ifprop: QHPRODS
<OCCURNUM>		NEWEXP	ifprop: QHPRODS
<OTHER2>		NEWEXP	ifprop: QHPRODS
<PLATIS>		NEWEXP	ifprop: QHPRODS
<TELLABT>		NEWEXP	ifprop: QHPRODS
<TYPE2>		NEWEXP	ifprop: QHPRODS
<TYPIS>		NEWEXP	ifprop: QHPRODS
<VALIS>		NEWEXP	ifprop: QHPRODS
<WHAT2FORM>		NEWEXP	ifprop: QHPRODS
<WHATFORM>		NEWEXP	ifprop: QHPRODS
<WHEREFORM>		NEWEXP	ifprop: QHPRODS
<WHEREITEM>		NEWEXP	ifprop: QHPRODS
<WHOSE2FORM>		NEWEXP	ifprop: QHPRODS
<WHOSEFORM>		NEWEXP	ifprop: QHPRODS
ADDH	88	HASHER	expr: (ARGS NEWVAL)
ADDIS	257	TOPELVEL	expr: (SN)
AFTERSYSOUTFORMS		FORK	ADDVARS
ALIAS		NEWEXP	ifprop: PRINFORMS
ANDHACK	103	INTERP	expr: (CONDITIONS ACTIONS EV)
APPLYRULE	104	INTERP	expr: (RULENAME PREBIND)
ARRLOC	75	FORK	expr: (ARR)
ASSERT	117	MANIPULATE	expr: (ARGLIST NODENAME)
ASSERTION		NEWEXP	Set Variable
ASSRPRINT	148	NEWEXP	expr: (PRINSPEC)
AUXINTERPOL	222	PLAT	expr: (PT1 PT2 DELTA)
BEARING		ORACLE	ifprop: ORACLE
BEARING		ORACLE	ifprop: ORTYPE
BEARING	181	ORACLE	expr: (SITE)
BEEP	231	QH	expr: NIL
BEYONDINTEREST	128	MSGMTR	expr: (TXT)
BKDSPBUF	28	DSPLA	expr: (X)
BLFN	5	CONFIDENCE	expr: (BNODE)
BLOCKED-FROM		NEWEXP	ifprop: PRINFORMS

BLOCKER		RULES	prop: CONDITIONS
BLOCKER		RULES	prop: ACTIONS
BLOCKER		RULES	prop: CONF
BMEAS	4	CONFIDENCE	expr: (BBOX)
BUMP	118	MANIPULATE	expr: (L)
CASSERT	119	MANIPULATE	expr: (SPEC VAL)
CENTROID	223	PLAT	expr: (VERTEXLIST)
CHANGECON	149	NEWEXP	expr: (RLNME1)
CKCONFIGURATION	258	TOPLEVEL	expr: NIL
CLASS		NEWEXP	ifprop: PRINFORMS
CLOSE-POPUP		RULES	prop: CONDITIONS
CLOSE-POPUP		RULES	prop: ACTIONS
CLOSE-POPUP		RULES	prop: CONF
CONFIDBLOCK		CONFIDENCE	BLOCKS
CONSTRUCT	105	INTERP	expr: (ACTIONS EV COUNT)
CONTACT		NEWEXP	ifprop: PRINFORMS
COURSE		NEWEXP	ifprop: PRINFORMS
COURSE		ORACLE	ifprop: ORACLE
COURSE		ORACLE	ifprop: ORTYPE
COURSE	211	ORACLE	expr: (SITE)
COURSE-CHANGED		RULES	prop: CONDITIONS
COURSE-CHANGED		RULES	prop: ACTIONS
COURSE-CHANGED		RULES	prop: CONF
COURSEFROM		NEWEXP	ifprop: PRINFORMS
COURSEFROM		ORACLE	ifprop: ORACLE
COURSEFROM		ORACLE	ifprop: ORTYPE
COURSEFROM	213	ORACLE	expr: (POS1 POS2)
CREATECONTACT		RULES	prop: CONDITIONS
CREATECONTACT		RULES	prop: ACTIONS
CREATECONTACT		RULES	prop: CONF
CREATEDETECT		RULES	prop: CONDITIONS
CREATEDETECT		RULES	prop: ACTIONS
CREATEDETECT		RULES	prop: CONF
CREATEPLAT		RULES	prop: CONDITIONS
CREATEPLAT		RULES	prop: ACTIONS
CREATEPLAT		RULES	prop: CONF
CREATH	89	HASHER	expr: (SIZE)
CROSSBOUNDARY	188	ORACLE	expr: (PT1 PT2 POLY)
CROSSLINES	191	ORACLE	expr: (A B P Q)
CROSSPATHS		NEWEXP	ifprop: PRINFORMS
CROSSPATHS		ORACLE	ifprop: ORACLE
CROSSPATHS		ORACLE	ifprop: ORTYPE
CROSSPATHS	197	ORACLE	expr: (S1 S2 T1 T2)
CRUNCH	18	DSPLA	expr: (X)
CURTIME		MSGMTR	Saved Variable
DECSAMEDIGITS	29	DSPLA	expr: (X)
DEFINEPD	236	RULES	expr: NIL
DENY	120	MANIPULATE	fexpr*: L
DESCRIBMSG	129	MSGMTR	expr: (TXT)
DETECTION		NEWEXP	ifprop: PRINFORMS
DIRECTION	209	ORACLE	expr: (LAT1 LON1 LAT2 LON2)
DISPCHECK	130	MSGMTR	expr: (NAME)
DISPLAY	131	MSGMTR	expr: (PLATNAME LAT LON TIME)
DISPLOB	132	MSGMTR	expr: (PNAME SPOS DPOS TIME)
DISPMARK	133	MSGMTR	expr: (NAME)
DISSIMILAR		NEWEXP	ifprop: PRINFORMS
DISSIMILPLAT	200	ORACLE	expr: (PLAT1 PLAT2)

DISTANCE	184	ORACLE	expr: (LAT1 LON1 LAT2 LON2)
DISTANT-POPUP		RULES	prop: CONDITIONS
DISTANT-POPUP		RULES	prop: ACTIONS
DISTANT-POPUP		RULES	prop: CONF
DISTOLINE	185	ORACLE	expr: (X Y X1 Y1 X2 Y2)
DLFN	8	CONFIDENCE	expr: (DNODE)
DMEAS	7	CONFIDENCE	expr: (DBOX)
DSPADDINC	23	DSPLA	expr: (NAME LAT LON TIME)
DSPADDINCS	41	DSPLA	expr: (NAME INCLST)
DSPADDTRH	22	DSPLA	expr: (NAME ID TYPE)
DSPCHGTRH	42	DSPLA	expr: (NAME ID TYPE)
DSPCMD	14	DSPLA	expr: (CMD WAITFLG)
DSPCNVRT	19	DSPLA	expr: (X)
DSPERASE	35	DSPLA	expr: NIL
DSPEXCH	36	DSPLA	expr: (NAME)
DSPEXCHBUF		DSPLA	Set Variable
DSPEXCHEMP		DSPLA	Set Variable
DSPEXP	172	NEWEXP	expr: (BOX)
DSPGRAB	24	DSPLA	expr: (TTYNO)
DSPINIT	15	DSPLA	expr: NIL
DSPLAYFLG		MSGMTR	Saved Variable
DSPMAP	37	DSPLA	expr: NIL
DSPNOMAP	38	DSPLA	expr: NIL
DSPNOWAITFLG		FORK	Set Variable
DSPNUMAT	20	DSPLA	expr: (X)
DSPQUIET	27	DSPLA	expr: NIL
DSPRELD	25	DSPLA	expr: NIL
DSPSAVE	43	DSPLA	expr: NIL
DSPSTAT	40	DSPLA	expr: NIL
DSPTOP	39	DSPLA	expr: (WAITFLG)
DSPTTY	26	DSPLA	expr: NIL
DSPTTYSTR	31	DSPLA	expr: NIL
DUALFLG		TOPLEVEL	Saved Variable
DULLREL		NEWEXP	Set Variable
EMITTER		NEWEXP	ifprop: PRINFORMS
ENDSTREAM	240	STREAM	expr: (S)
ESTIMATE	216	PLAT	expr: (SITE1 SITE2 GAP)
EWMSG	134	MSGMTR	expr: (TXT EXTFLG)
EXLOOP	259	TOPLEVEL	expr: NIL
EXPLAIN	150	NEWEXP	expr: NIL
EXPLAINFLAG		NEWEXP	Saved Variable
FANCYPROD	238	RULES	expr: (PRO)
FASTER-THAN-A-MERCHANT		RULES	prop: CONDITIONS
FASTER-THAN-A-MERCHANT		RULES	prop: ACTIONS
FASTER-THAN-A-MERCHANT		RULES	prop: CONF
FASTHAK	90	HASHER	expr: NIL
FIRST-SIGHTING		NEWEXP	ifprop: PRINFORMS
FIRST-VIEW		RULES	prop: CONDITIONS
FIRST-VIEW		RULES	prop: ACTIONS
FIRST-VIEW		RULES	prop: CONF
FIXLONG	225	PLAT	expr: (X)
FKACS	54	FORK	expr: NIL
FKACSRETURN	55	FORK	expr: (ARRAY)
FKARRADR	73	FORK	expr: (FKARRNAME FKINDEX FKNWORDS)

FKARRAY	62	FORK	fexpr: (FKA FKTYPE FKSIZE FKSIZE2)
FKARRAYBLOCK		FORK	BLOCKS
FKARRAYP	68	FORK	expr: (A)
FKARRAYSIZE	69	FORK	expr: (A)
FKARRAYTYPE	71	FORK	expr: (A)
FKBCHECK	72	FORK	expr: (N LO HI)
FKCALL	48	FORK	fexpr*: FKCX
FKCALLBLOCK		FORK	BLOCKS
FKCALLERR	58	FORK	expr: (FKCID)
FKCATYPE	49	FORK	expr: (FKID)
FKCORGET	63	FORK	expr: (SIZE)
FKDDT	47	FORK	expr: (DDTFILE)
FKDDT		FORK	prop: MACRO
FKDDT_		FORK	prop: MACRO
FKELT	64	FORK	fexpr: (FKELT!A FKELT!N FKELT!WORDS)
FKELTI	65	FORK	fexpr: (FKELTI!A FKELTI!N FKELTI!WORDS)
FKELTR	66	FORK	fexpr: (FKELTRI!A FKELTRI!N FKELTRI!WORDS)
FKFLOAT	74	FORK	expr: (ADR)
FKHALT		FORK	prop: MACRO
FKHNDL		FORK	prop: MACRO
FKHT		FORK	prop: MACRO
FKHT_		FORK	prop: MACRO
FKIDPB		FORK	prop: MACRO
FKINIT	44	FORK	fexpr: (PROGRAM)
FKJFN		FORK	prop: MACRO
FKJSYS	85	FORK	expr: (FKJSYSNO ARG1 ARG2 ARG3 ARG4 ARG5)
FKJSYSARG	86	FORK	expr: (X)
FKJSYSBLOCK		FORK	BLOCKS
FKKILL	45	FORK	expr: NIL
FKPROG		FORK	prop: MACRO
FKRACS		FORK	prop: MACRO
FKRTN	56	FORK	expr: (TYPE A N)
FKSACS		FORK	prop: MACRO
FKSAVE	46	FORK	expr: (FILE)
FKSETA	67	FORK	fexpr: (FKARRY FKINDEX FKEXPR)
FKSETVAL	79	FORK	fexpr: (FKADR FKBIAS FKVAL)
FKSHR		FORK	prop: MACRO
FKSR	50	FORK	expr: (A I STR)
FKSW	59	FORK	expr: (FKHNDL I FKNOWAITFLG)
FKSYM	80	FORK	expr: (ID FKHT NOBREAK)
FKSYMACS		FORK	prop: MACRO
FKSYMBLOCK		FORK	BLOCKS
FKSYMP	82	FORK	expr: (ID)
FKSYMPUT	81	FORK	expr: (FKHT ID V)
FKTIME	84	FORK	expr: (FKEXPR)
FKTTYSET	61	FORK	expr: (BOOL)
FKVAL	76	FORK	fexpr: (FKADR FKBIAS FKWORDS)
FKVALAT	21	DSPLA	fexpr: (ID BIAS NVALS)
FKVALI	78	FORK	fexpr: (FKADR FKBIAS FKWORDS FKREAL)
FKVALR	77	FORK	fexpr: (FKADR FKBIAS FKWORDS)
FKWAIT	87	FORK	expr: (FKHNDL)
FKX	60	FORK	fexpr: (FKCX)
FREEZE	241	STREAM	expr: NIL
FREEZEFLG		STREAM	Set Variable

FREEZELST		STREAM	Set Variable
FROM-PORT		NEWEXP	ifprop: PRINFORMS
GAMF	151	NEWEXP	expr: (WLK OVERRIDE)
GETATT	215	PLAT	expr: (REL NAME)
GETATTB	178	ORACLE	expr: (REL NODE)
GETCON	1	CONFIDENCE	expr: (SOMAST)
GETH	91	HASHER	expr: (ARGS)
GETMARK	2	CONFIDENCE	expr: (NODE)
GETMB	3	CONFIDENCE	expr: (BAST)
GETMD	6	CONFIDENCE	expr: (DAST)
GETMRVAL	250	STREAM	expr: (X COPYFLG)
GETPOINT	224	PLAT	expr: (POS BEAR RANGE)
GETPULSAR	106	INTERP	expr: (NODE)
GETRADIX50	83	FORK	expr: (S)
GETSH	92	HASHER	expr: (ARGS)
GETSTRIP	93	HASHER	expr: (ARGS)
GETUPLE	121	MANIPULATE	expr: (ASSER)
GLOBALVARS		FORK	ADDVARS
GRATEK	11	DSPLA	expr: NIL
GRAZE		NEWEXP	ifprop: PRINFORMS
GRAZE		ORACLE	ifprop: ORACLE
GRAZE		ORACLE	ifprop: ORTYPE
GRAZE	206	ORACLE	expr: (S1 S2 T1 T2)
GREATER-THAN		NEWEXP	ifprop: PRINFORMS
GREATER-THAN		ORACLE	ifprop: ORACLE
GREATER-THAN		ORACLE	ifprop: ORTYPE
GREATER-THAN	179	ORACLE	expr: (Q1 Q2)
GREATESTPROB	135	MSGMTR	expr: (POSLIST)
HLPEXPLN	152	NEWEXP	expr: NIL
ID		NEWEXP	ifprop: PRINFORMS
ID-AMPLIFY		NEWEXP	ifprop: PRINFORMS
ID-LANE		RULES	prop: CONDITIONS
ID-LANE		RULES	prop: ACTIONS
ID-LANE		RULES	prop: CONF
IDENT	136	MSGMTR	expr: (NAME)
IMPLIESASRT	153	NEWEXP	expr: (NODE)
IN-LANE		NEWEXP	ifprop: PRINFORMS
IN-LANE		ORACLE	ifprop: ORACLE
IN-LANE		ORACLE	ifprop: ORTYPE
IN-LANE	176	ORACLE	expr: (MLANE POS)
INCLUDEPLAT	260	TOPLEVEL	expr: (PNE)
INHERIT		RULES	prop: CONDITIONS
INHERIT		RULES	prop: ACTIONS
INHERIT		RULES	prop: CONF
INLANE	186	ORACLE	expr: (X Y LANE)
INSIDE		NEWEXP	ifprop: PRINFORMS
INSIDE		ORACLE	ifprop: ORACLE
INSIDE		ORACLE	ifprop: ORTYPE
INSIDE	177	ORACLE	expr: (POS STORM)
INSIDE-A-MERCHANTLANE		NEWEXP	ifprop: PRINFORMS
INSIDE-A-STORM		RULES	prop: CONDITIONS
INSIDE-A-STORM		RULES	prop: ACTIONS
INSIDE-A-STORM		RULES	prop: CONF
INTERIOR	183	ORACLE	expr: (OLAT OLON POLYGON)
INTERPOLABLE	137	MSGMTR	expr: (TXT)
JUGGLE	154	NEWEXP	expr: (PAIR INSERTITEM)

JUSTBUILD	107	INTERP	expr: (SPEC EV NUMBER)
LAND-DIST		NEWEXP	ifprop: PRINFORMS
LANERANGE	195	ORACLE	expr: (ALAT ALON BLAT BLON CLAT CLON)
LESS-THAN		NEWEXP	ifprop: PRINFORMS
LESS-THAN		ORACLE	ifprop: ORACLE
LESS-THAN		ORACLE	ifprop: ORTYPE
LESS-THAN	180	ORACLE	expr: (Q1 Q2)
LINERead	239	RULES	expr: NIL
LINPOLY	187	ORACLE	expr: (PT1 PT2 POLY)
LOC-TIME	203	ORACLE	expr: (S)
LOCATION		NEWEXP	ifprop: PRINFORMS
LOCATION	198	ORACLE	expr: (S)
LOCH	94	HASHER	expr: (ARGS PUTFLG)
M	12	DSPLA	fexpr*: L
MAKEPD	237	RULES	expr: (NAM CO AC TRUST)
MAKEPRINT	155	NEWEXP	expr: (RELN)
MAPH	95	HASHER	expr: (ARY ARYSZ ARYFN)
MAPRETALIST		STREAM	Set Variable
MAPRETDO	245	STREAM	expr: (SELT AI)
MAPRETRIEVE	244	STREAM	expr: (MAPRETX MAPRETINFO MAPRETFN)
MAPSTREAM	242	STREAM	expr: (MAPSTREAMX MAPSTREAMINFO MAPSTREAMFN)
MARKOFF	9	CONFIDENCE	expr: (NODE)
MARKON	10	CONFIDENCE	expr: (NODE MARK)
MESSAGE1	108	INTERP	expr: (SPECLIST)
MATCH-PLAT		RULES	prop: CONDITIONS
MATCH-PLAT		RULES	prop: ACTIONS
MATCH-PLAT		RULES	prop: CONF
MATCHER	122	MANIPULATE	expr: (L1 L2)
MAXSHIPSPEED		ORACLE	Saved Variable
MAYBE	123	MANIPULATE	fexpr*: L
MEDIUM	138	MSGMTR	expr: (NAME)
MEDIUM		NEWEXP	ifprop: PRINFORMS
MELD	139	MSGMTR	expr: (ID MED)
MEMDENSITY	96	HASHER	expr: NIL
MEMFACTOR		HASHER	Saved Variable
MEMLIMIT		HASHER	Saved Variable
MEMORY		HASHER	Saved Variable
MEMSAVE	156	NEWEXP	expr: (FEE)
MEMSIZE		HASHER	Saved Variable
MEMTEST	97	HASHER	expr: (X Y)
MERCHANTLANE		NEWEXP	ifprop: PRINFORMS
MIDP	140	MSGMTR	expr: (P1 P2)
MODE		NEWEXP	ifprop: PRINFORMS
MODIFIER	157	NEWEXP	expr: NIL
MONTEK	13	DSPLA	expr: NIL
MSGFILE		MSGMTR	Saved Variable
MSGMTP	141	MSGMTR	expr: NIL
NEAREST	217	PLAT	expr: (PT LST)
NEWHASH	98	HASHER	expr: NIL
NEWSTREAM	243	STREAM	expr: NIL
NEWSYM	142	MSGMTR	expr: (NAME)
NEWVALOBJ	158	NEWEXP	expr: (ARRT)
NEXTH	99	HASHER	expr: (LOC ARG)
NICEANSWER	159	NEWEXP	expr: (ANS1)
NOFORK	57	FORK	expr: NIL
NOT-FIRST		NEWEXP	ifprop: PRINFORMS

NOT-FIRST-SIGHTING		RULES	prop: CONDITIONS
NOT-FIRST-SIGHTING		RULES	prop: ACTIONS
NOT-FIRST-SIGHTING		RULES	prop: CONF
NOT-KNOWN-COMBATANT		RULES	prop: CONDITIONS
NOT-KNOWN-COMBATANT		RULES	prop: ACTIONS
NOT-KNOWN-COMBATANT		RULES	prop: CONF
NOT-LAST		NEWEXP	ifprop: PRINFORMS
NOT-LAST-SIGHTING		RULES	prop: CONDITIONS
NOT-LAST-SIGHTING		RULES	prop: ACTIONS
NOT-LAST-SIGHTING		RULES	prop: CONF
NOTHACK	109	INTERP	expr: (CONDITIONS ACTIONS EV)
OCCURPRINT	160	NEWEXP	expr: (TIMES NODE)
OCTSAMEDIGITS	30	DSPLA	expr: (X)
ONEPOINT	218	PLAT	expr: (NODE GAP)
OPSIDES	192	ORACLE	expr: (A B P Q)
ORACLEHACK	110	INTERP	expr: (SPEC)
ORACLES		ORACLE	Saved Variable
ORBUILD	111	INTERP	expr: (SPEC EV)
ORHACK	112	INTERP	expr: (CONDITIONS ACTIONS EV)
OUTSIDE-ALL-LANES		RULES	prop: CONDITIONS
OUTSIDE-ALL-LANES		RULES	prop: ACTIONS
OUTSIDE-ALL-LANES		RULES	prop: CONF
OWNMSG	143	MSGMTR	expr: (TXT)
OWNPOS	144	MSGMTR	expr: (TIME)
OWNSHIP		MSGMTR	Saved Variable
OWNSHIP		NEWEXP	ifprop: PRINFORMS
PARTING	261	TOPELVEL	expr: NIL
PATROL		NEWEXP	ifprop: PRINFORMS
PLATFORM		NEWEXP	ifprop: PRINFORMS
PLATPOS	219	PLAT	expr: (PLAT TIME)
POSITION		NEWEXP	ifprop: PRINFORMS
POSS-REPORT	199	ORACLE	expr: (S1 S2 PATROL)
POSS-RPT		RULES	prop: CONDITIONS
POSS-RPT		RULES	prop: ACTIONS
POSS-RPT		RULES	prop: CONF
POSSIBLE-REPORT		NEWEXP	ifprop: PRINFORMS
PQ	226	QH	fexpr*: L
PREDECESSOR		ORACLE	ifprop: ORACLE
PREDECESSOR		ORACLE	ifprop: ORTYPE
PREDECESSOR	208	ORACLE	expr: (SITE)
PREDICTPOS	220	PLAT	expr: (NODELIST TIME)
PREHASH	100	HASHER	expr: (L)
PREPALIST	248	STREAM	expr: (CON ASS ALIST)
PRETTYANS	161	NEWEXP	expr: (ANSLST)
PRETTYASSR	162	NEWEXP	expr: (NODE FORMAT OVERCONF)
PRINCHAR	16	DSPLA	expr: (CODE)
PRINTRULEASSR	163	NEWEXP	expr: (RULEASSRTS)
PRODUCTIONS		RULES	Saved Variable
PULSAR	253	STREAM	expr: NIL
PULSE	254	STREAM	expr: (PULSAR)
PUTH	101	HASHER	expr: (ARGS AVAL)
PUTSH	102	HASHER	expr: (ARGS AVAL)
PUTSTREAM	255	STREAM	expr: (S X)
PUTTYP		FORK	prop: MACRO
QHASK	230	QH	expr: (INBUF)
QHCLEAR	227	QH	expr: NIL
QHFOLLOW	233	QH	expr: (LL BUFPTR QHMATCH)

QHGET		QH	MACROS
QHLIST	229	QH	expr: (PTR)
QHMAKE	228	QH	expr: (QHMAKEX QHMAKEY SHOWFLG)
QHPREP	234	QH	expr: (FOCUS QHLST SHOWFLG STK)
QHPUT		QH	MACROS
QHSHOW	235	QH	expr: (L)
QHTAKE	232	QH	fexpr*: L
RADAR-MODE		NEWEXP	ifprop: PRINFORMS
RANGE		NEWEXP	ifprop: PRINFORMS
RANGE		ORACLE	ifprop: ORACLE
RANGE		ORACLE	ifprop: ORTYPE
RANGE	210	ORACLE	expr: (SITE)
REACHABLE		RULES	prop: CONDITIONS
REACHABLE		RULES	prop: ACTIONS
REACHABLE		RULES	prop: CONF
REACHABLE-BY-A-COMBATANT			
		NEWEXP	ifprop: PRINFORMS
RECAPCONCS	164	NEWEXP	expr: NIL
RELATIONS		NEWEXP	Saved Variable
RESOUT	165	NEWEXP	expr: NIL
RESULTLIST		TOPLEVEL	Saved Variable
RESULTPRINTER	166	NEWEXP	expr: (RES1)
RETPULSED0	246	STREAM	expr: (SELTAI)
RETRIEVER	124	MANIPULATE	expr: (SPEC)
RETRIEVES	247	STREAM	expr: (AT OBJ VAL SEL)
RETSTREAM	249	STREAM	expr: (C)
RETVARs	125	MANIPULATE	expr: (SPEC)
ROTSENSE	193	ORACLE	expr: (A B C)
ROUGHLY-THE-SAME-COURSE-AS			
		NEWEXP	ifprop: PRINFORMS
ROUGHLY-THE-SAME-COURSE-AS			
		ORACLE	ifprop: ORACLE
ROUGHLY-THE-SAME-COURSE-AS			
		ORACLE	ifprop: ORTYPE
ROUGHLY-THE-SAME-COURSE-AS			
	175	ORACLE	expr: (Q1 Q2)
ROUGHLY-THE-SAME-SPEED-AS			
		NEWEXP	ifprop: PRINFORMS
ROUGHLY-THE-SAME-SPEED-AS			
		ORACLE	ifprop: ORACLE
ROUGHLY-THE-SAME-SPEED-AS			
		ORACLE	ifprop: ORTYPE
ROUGHLY-THE-SAME-SPEED-AS			
	174	ORACLE	expr: (Q1 Q2)
RULE		NEWEXP	Set Variable
RULEXP	167	NEWEXP	expr: (RULE NODE)
SAILARG	52	FORK	expr: (FKARG FKHT)
SAILARRAYSIZE	70	FORK	expr: (A)
SAILCALL	51	FORK	fexpr*: FKCX
SAILSTRING	53	FORK	expr: (STRING)
SAME-AS		NEWEXP	ifprop: PRINFORMS
SAME-AS		ORACLE	ifprop: ORACLE
SAME-AS		ORACLE	ifprop: ORTYPE
SAME-AS	173	ORACLE	expr: (W U)
SAVEPULSAR	113	INTERP	expr: (NODE)
SCRATCHFIVE		DSPLA	Set Variable
SCRATCHTEN		DSPLA	Set Variable

SENSORANGE		MSGMTR	Saved Variable
SENSORMSG	145	MSGMTR	expr: (TXT)
SERT	126	MANIPULATE	expr: (SPEC NODENAME)
SIGHTING		NEWEXP	ifprop: PRINFORMS
SIMPLY-REACHABLE		RULES	prop: CONDITIONS
SIMPLY-REACHABLE		RULES	prop: ACTIONS
SIMPLY-REACHABLE		RULES	prop: CONF
SIMPLY-WITHIN-REACH		NEWEXP	ifprop: PRINFORMS
SLOWER-THAN-A-MERCHANT		RULES	prop: CONDITIONS
SLOWER-THAN-A-MERCHANT		RULES	prop: ACTIONS
SLOWER-THAN-A-MERCHANT		RULES	prop: CONF
SMALL-CRAFT1		RULES	prop: CONDITIONS
SMALL-CRAFT1		RULES	prop: ACTIONS
SMALL-CRAFT1		RULES	prop: CONF
SMALL-CRAFT2		RULES	prop: CONDITIONS
SMALL-CRAFT2		RULES	prop: ACTIONS
SMALL-CRAFT2		RULES	prop: CONF
SMALL-CRAFT3		RULES	prop: CONDITIONS
SMALL-CRAFT3		RULES	prop: ACTIONS
SMALL-CRAFT3		RULES	prop: CONF
SMALL-CRAFT4		RULES	prop: CONDITIONS
SMALL-CRAFT4		RULES	prop: ACTIONS
SMALL-CRAFT4		RULES	prop: CONF
SMALL-CRAFT5		RULES	prop: CONDITIONS
SMALL-CRAFT5		RULES	prop: ACTIONS
SMALL-CRAFT5		RULES	prop: CONF
SMALL-CRAFT6		RULES	prop: CONDITIONS
SMALL-CRAFT6		RULES	prop: ACTIONS
SMALL-CRAFT6		RULES	prop: CONF
SMALL-CRAFT9		RULES	prop: CONDITIONS
SMALL-CRAFT9		RULES	prop: ACTIONS
SMALL-CRAFT9		RULES	prop: CONF
SMALLNUMB		NEWEXP	Set Variable
SOMELINESEG	189	ORACLE	expr: (SOMELINESEGX SOMELINESEGFN)
SOMEPUSE	251	STREAM	expr: (PULSAR PULSARDATA SOMEPUSEFN)
SOURCE		NEWEXP	ifprop: PRINFORMS
SPAN	221	PLAT	expr: (L1 L2)
SPEED		NEWEXP	ifprop: PRINFORMS
SPEED		ORACLE	ifprop: ORACLE
SPEED		ORACLE	ifprop: ORTYPE
SPEED	182	ORACLE	expr: (SITE)
SPEED-CHANGED		RULES	prop: CONDITIONS
SPEED-CHANGED		RULES	prop: ACTIONS
SPEED-CHANGED		RULES	prop: CONF
SPEEDAUX	212	ORACLE	expr: (T1 T2 DIST)
SPEEDFROM		NEWEXP	ifprop: PRINFORMS
SPEEDFROM		ORACLE	ifprop: ORACLE
SPEEDFROM		ORACLE	ifprop: ORTYPE
SPEEDFROM	214	ORACLE	expr: (POS1 T1 POS2 T2)
SPEEDM	205	ORACLE	expr: (T1 T2 DIST)
STAMMER	262	TOPELVEL	expr: NIL
STARTUP	263	TOPELVEL	expr: NIL
STATE	127	MANIPULATE	fexpr*: L
STATES		NEWEXP	Saved Variable

STRENGTH		NEWEXP	ifprop: PRINFORMS
STRIPSTREAM	252	STREAM	expr: (S)
STUFFLN	264	TOPLEVEL	expr: (MLN)
SUBTEND	194	ORACLE	expr: (LAT1 LON1 LAT2 LON2)
SUCCESSOR		NEWEXP	ifprop: PRINFORMS
SUCCESSOR		ORACLE	ifprop: ORACLE
SUCCESSOR		ORACLE	ifprop: ORTYPE
SUCCESSOR	207	ORACLE	expr: (SITE)
SWEeper	114	INTERP	expr: (CONDITIONS ACTIONS EV)
SWR		NEWEXP	ifprop: PRINFORMS
SWR		ORACLE	ifprop: ORACLE
SWR		ORACLE	ifprop: ORTYPE
SWR	204	ORACLE	expr: (LT1 T1 LT2 T2)
TEKCOM	32	DSPLA	expr: (STR)
TEKTEST	33	DSPLA	expr: NIL
TEKWAIT	34	DSPLA	expr: NIL
TO-PORT		NEWEXP	ifprop: PRINFORMS
TOS		NEWEXP	ifprop: PRINFORMS
TRACKINPOLY	190	ORACLE	expr: (TRACK POLY)
TWO-PLACE	146	MSGMTR	expr: (X)
TYPE		NEWEXP	ifprop: PRINFORMS
UNCRUNCH	17	DSPLA	expr: (NUM)
UNFREEZE	256	STREAM	expr: NIL
UNLESSHACK	115	INTERP	expr: (CONDITIONS ACTIONS EV)
VAR?	116	INTERP	expr: (Q)
VDRELS		INTERP	Set Variable
WAITER	265	TOPLEVEL	expr: NIL
WEATHERMSG	147	MSGMTR	expr: (TXT)
WELCOME	266	TOPLEVEL	expr: NIL
WENT-AFTER		NEWEXP	ifprop: PRINFORMS
WENT-AFTER		ORACLE	ifprop: ORACLE
WENT-AFTER		ORACLE	ifprop: ORTYPE
WENT-AFTER	202	ORACLE	expr: (S1 T1 S2 T2 S3 T3 S4 T4)
WENT-BEFORE		NEWEXP	ifprop: PRINFORMS
WENT-BEFORE		ORACLE	ifprop: ORACLE
WENT-BEFORE		ORACLE	ifprop: ORTYPE
WENT-BEFORE	201	ORACLE	expr: (S1 T1 S2 T2 S3 T3 S4 T4)
WHAT2FORMFN	168	NEWEXP	expr: (PL)
WHATFORMFN	169	NEWEXP	expr: (REL OBJ)
WHOSE2FORMFN	170	NEWEXP	expr: (VAL REL)
WITHIN-REACH		NEWEXP	ifprop: PRINFORMS
WITHINR	196	ORACLE	fexpr*: L
YESNO	171	NEWEXP	expr: (ASSRSPEC)
carriagereturn		NEWEXP	Saved Variable

(FILECREATED "24-Jul-79 13:44:49" <RBECHTAL>CONFIDENCE..23 7975

changes to: BMEAS DMEAS

previous date: "23-Jul-79 18:42:31" <RBECHTAL>CONFIDENCE..22)

(PRETTYCOMPRINT CONFIDENCECOMS)

(RPAQQ CONFIDENCECOMS ((FNS * CONFIDENCEFNS)
(BLOCKS * CONFIDENCEBLOCKS)))

(RPAQQ CONFIDENCEFNS (GETCON GETMARK GETMB BMEAS BLFN GETMD DMEAS DLFN
MARKOFF MARKON))

(DEFINEQ

[1]

(GETCON
[LAMBDA (SOMAST)

(* edited:
"19-Jul-79 12:49")

(* GETCON computes the confidence in an assertion,
which is defined as the measure of belief in the
assertion less the measure of disbelief in the
assertion. GETCON will also accept a list of
assertions.)

(COND
((NULL SOMAST)
0.0)
((ATOM SOMAST)
(FDIFFERENCE (GETMB SOMAST)
(GETMD SOMAST)))
((LISTP SOMAST)
(MAPCAR SOMAST (FUNCTION GETCON))

[2]

(GETMARK
[LAMBDA (NODE)
(GETPROP NODE (QUOTE SUPERMARK))

(* edited:
"23-Jul-79 16:44")

[3]

(GETMB
[LAMBDA (BAST)

(* edited:
"23-Jul-79 18:36")

(* GETMB calculates the measure of belief in an
assertion. If there is a derivation tree, the belief
derived through it is preferred to the belief stored
directly on the property list
(if any). GETMB gets the list of derivation boxes

and maps the function BMEAS overthem.
 When BMEAS is done, MBCOMB will hold the measure of
 belief (accumulated in accordance with the combining
 function developed for MYCIN))

```
(PROG ((HNDL (GETPROP BAST (QUOTE DERIVE*)))
      (MBCOMB 0.0))
  (COND
    ((EQ (GETMARK BAST)
         (QUOTE POS))
     (RETURN 0.0))
    ((EQ (GETMARK BAST)
         (QUOTE NEG))
     (RETURN 1.0))
    (HNDL (MAPC HNDL (FUNCTION BMEAS))
      (RETURN MBCOMB))
    (T (RETURN (GETPROP BAST (QUOTE MB))
```

[4]

(BMEAS
 [LAMBDA (BBOX)

(* edited:
 "24-Jul-79 13:41")

(* BMEAS operates on a single derivation box.
 If the box provides negative evidence it is ignored
 (it will be counted for the measure of disbelief).
 Otherwise the functions BLFN and DLFN are mapped
 over the assertion entries in the box.
 This results in BMEASANS being set to the minimum of
 the belief measures for each assertion in the box,
 while DMEASANS is set to the maximum of the
 disbelief measures. The difference between BMEASANS
 and DMEASANS, if positive, is multiplied by the rule
 confidence and combined with the measures produced
 by the other boxes. If the difference is negative,
 this box is ignored.)

```
(PROG ((BASTLST (CDR BBOX))
      (RULECON (GETPROP (CAR BBOX)
                        (QUOTE CONF))))
  (BMEASANS 1.0)
  (DMEASANS 0.0))
(COND
  ((MINUSP RULECON)
   (RETURN)))
(RESETLST [RESETSAVE (MARKON BAST (QUOTE POS))
                (QUOTE (AND (MARKOFF BAST]
                (MAPC BASTLST (FUNCTION BLFN))
                (MAPC BASTLST (FUNCTION DLFN)))
(COND
  ((FGREATERP BMEASANS DMEASANS)
   (SETQ BMEASANS (FTIMES RULECON (FDIFFERENCE BMEASANS
                                                  DMEASANS)))
   (SETQ MBCOMB (FDIFFERENCE (FPLUS BMEASANS MBCOMB)
```

(FTIMES BMEASANS MBCOMB)]

[5]

(BLFN
[LAMBDA (BNODE)

(* edited:
"23-Jul-79 18:38")

(* BLFN looks at a single node contained in a derivation box. If the node is satisfying a negation condition, the measure of belief for use in confidence calculation is taken to be the measure of disbelief in the assertion. If the node is satisfying an unless condition, the measure of belief used is 1.0 if the confidence in the node is 0.0 or less, otherwise the measure of belief is zero. For AND and OR conditions, BLFN just uses the measure of belief in the assertion. Having acquired a usable measure of belief, BLFN then tests this against BMEASANS (the minimal MB to this point), and sets BMEASANS to the minimum of these two.)

(PROG ((BNCON 0.0))

(* This sets up BNCON
with a floating number
box)

```

(COND
  ((LISTP BNODE)
    (SELECTQ (CAR BNODE)
      [NOT (SETQ BNCON (GETMD (CADR BNODE)
        [UNLESS (PROGN (SETQ BNCON (GETCON (CADR BNODE)))
          (COND
            ((GREATERP BNCON 0.0)
              (SETQ BNCON 0.0))
            (T (SETQ BNCON 1.0))
          )
        (SETQ BNCON 0.0)))
      (T (SETQ BNCON (GETMB BNODE)
        (SETQ BMEASANS (MIN BNCON BMEASANS))

```

[6]

(GETMD
[LAMBDA (DAST)

(* edited:
"23-Jul-79 18:39")

(* GETMD corresponds to GETMB for measures of disbelief. It's used the same way, but calls its own subsidiary functions, DMEAS and DLFN.)

```

(PROG ((DNDL (GETPROP DAST (QUOTE DERIVE*)))
  (MDCOMB 0.0))
(COND
  ((EQ (GETMARK DAST)
    (QUOTE POS))
    (RETURN 1.0))

```

```

((EQ (GETMARK DAST)
      (QUOTE NEG))
 (RETURN 0.0))
(DNDL (MAPC DNDL (FUNCTION DMEAS))
      (RETURN MDCOMB))
(T (RETURN (GETPROP DAST (QUOTE MD))

```

[7]

```

(DMEAS
 [LAMBDA (DBOX)

```

```

(* edited:
"24-Jul-79 13:44")
(* See BMEAS for a
description. Substitute
DLFN for BLFN, etc.)

```

```

(PROG ((DASTLST (CDR DBOX))
      (RULECON (GETPROP (CAR DBOX)
                        (QUOTE CONF)))
      (BMEASANS 1.0)
      (DMEASANS 0.0))
 (COND
  ((NOT (MINUSP RULECON))
   (RETURN)))
 (RESETLST [RESETSAVE (MARKON DAST (QUOTE NEG))
                (QUOTE (AND (MARKOFF DAST)
                          (MAPC DASTLST (FUNCTION BLFN))
                          (MAPC DASTLST (FUNCTION DLFN))))
 (COND
  ((FGREATERP BMEASANS DMEASANS)
   (SETQ DMEASANS (FTIMES RULECON (FDIFFERENCE DMEASANS
                                                  BMEASANS)))
   (* Negative by negative
      gives positive)
   (SETQ MDCOMB (FDIFFERENCE (FPLUS DMEASANS MDCOMB)
                              (FTIMES DMEASANS MDCOMB)))

```

[8]

```

(DLFN
 [LAMBDA (DNODE)

```

```

(* edited:
"23-Jul-79 18:41")
(* See BLFN for a
description of the
actions.)

```

```

(PROG ((DNCON 0.0))
 [COND
  ((LISTP DNODE)
   (SELECTQ (CAR DNODE)
    [NOT (SETQ DNCON (GETMB (CADR DNODE)
                          (UNLESS (PROGN (SETQ DNCON (GETCON (CADR DNODE)))
                                (COND
                                 ((FGREATERP DNCON 0.0)
                                  (SETQ DNCON 1.0))
                                 (T (SETQ DNCON 0.0))
                                (SETQ DNCON 0.0)))
    (T (SETQ DNCON (GETMD DNODE)
      (SETQ DMEASANS (MAX DNCON DMEASANS))

```

[9]

(MARKOFF
[LAMBDA (NODE)

(* edited:
"23-Jul-79 16:44")

(REMPROP NODE (QUOTE SUPERMARK))

[10]

(MARKON
[LAMBDA (NODE MARK)

(* edited:
"23-Jul-79 16:43")

(PUTPROP NODE (QUOTE SUPERMARK)
MARK])

)

(RPAQQ CONFIDENCEBLOCKS ((CONFIDEBLOCK GETCON GETMB BMEAS BLFN GETMD
DMEAS DLFN
(ENTRIES GETCON GETMB BMEAS BLFN
GETMD DMEAS DLFN)
(SPECVARS MBCOMB BAST BMEASANS
MDCOMB DAST DMEASANS)))

)

[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: CONFIDEBLOCK GETCON GETMB BMEAS BLFN GETMD DMEAS DLFN
(ENTRIES GETCON GETMB BMEAS BLFN GETMD DMEAS DLFN)
(SPECVARS MBCOMB BAST BMEASANS MDCOMB DAST DMEASANS))

]

(DECLARE: DONTCOPY
(FILEMAP (NIL (399 7481 (GETCON 411 . 923) (GETMARK 927 . 1060) (GETMB
1064 . 2079) (BMEAS 2083 . 3494) (BLFN 3498 . 4851) (GETMD 4855 . 5573)
(DMEAS 5577 . 6526) (DLFN 6530 . 7192) (MARKOFF 7196 . 7329) (MARKON
7333 . 7478)))))
STOP

(FILECREATED "10-Aug-79 16:19:32" <PMORRIS>DSPLA.LSP.88 10319

changes to: TEKTEST

previous date: " 1-Aug-79 18:01:40" <PMORRIS>DSPLA.LSP.87)

PRETTYCOMPRINT DSPLACOMS)

```

RPAQQ DSPLACOMS [(FNS * DSPLAFNS)
  (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVAR
    (ADDVARS (NLAMA M)
              (NLAML FKVALAT)
              (LAMA)))
  (VARS (SCRATCHTEN (QUOTE (0 0 0 0 0 0 0 0 0 0)))
        (SCRATCHFIVE (QUOTE (0 0 0 0 0)))
        (DSPEXCHBUF (CONCAT
          "
          (CHARACTER 0)))
          (DSPEXCHEMP (CONCAT
            "
            E      M      P      T      Y
            (CHARACTER 0]))

```

```

(RPAQQ DSPLAFNS (GRATEK M MONTEK DSPCMD DSPINIT PRINCHAR UNCRUNCH
  CRUNCH DSPCNVRT DSPNUMAT FKVALAT DSPADDTRH
  DSPADDINC DSPGRAB DSPRELD DSPTTY DSPQUIET
  BKDSPBUF DECSAMEDIGITS OCTSAMEDIGITS DSPTTYSTR
  TEKCOM TEKTEST TEKWAIT DSPERASE DSPEXCH DSPMAP
  DSPNOMAP DSPTOP DSPSTAT DSPADDINCS DSPCHGTRH
  DSPSAVE))

```

(DEFINEQ

[11]

```

(GRATEK
  [LAMBDA NIL

```

```

(* edited:
  " 1-Aug-79 17:48")

```

```

  (TEKCOM "WOR 33 H")
  (TEKCOM "GRA 3,33")
  (TEKCOM "SHR B")
  (PRINCHAR 29)
  (PRIN1 "BONE")
  (JSYS 60 31)
  (PRINCHAR 27)
  (PRINCHAR 12)
  (TERPRI)
  (DOBE])

```

[12]

```

(M
  [NLAMBDA L
    (NCONC DSPLAFNS L)
    (MAKEFILE (QUOTE DSPLA.LSP))]

```

[13]

(MONTEK
[LAMBDA NIL

(* edited:
"30-Jul-79 19:31")

(TEKCOM "MON 34"))

[14]

(DSPCMD
[LAMBDA (CMD WAITFLG)

(* edited:
"30-Jul-79 16:03")

(PROG (DSPNOWAITFLG)
(FKCALL ERASE SUBR))

(COND
(TEK4025 (GRATEK)))

(PROG ((DSPNOWAITFLG T))

[COND

((OR WAITFLG (EQP DSPTTYCODE 262143))

(SETQ DSPNOWAITFLG (GETTOPVAL (QUOTE DSPNOWAITFLG)

(FKCALL DSPLA SUBR CMD (NCHARS CMD))

(TERPRI))

(COND

(TEK4025 (MONTEK)))

[15]

(DSPINIT
[LAMBDA NIL
(SETQ DSPNOWAITFLG NIL)
(FKINIT DSPLIB)
(DSPQUIET)
(BKDSPBUF "1.3

1.0
NO
")

(* THIS "UNREADS" THE STRING, IE.
PLACES IT IN THE DISPLAY INPUT BUFFER, SO THAT IT
WILL BE READ BY THE FORTRAN SUBROUTINE DSPLAI)

(DSPCNVRT (DSPTTYSTR))
(FKSETVAL NTTY 1 (LIST DSPWORD1 DSPWORD2))
(FKCALL FRTEND SUBR)
(DSPTTY)
(TERPRI))

[16]

(PRINCHAR
[LAMBDA (CODE)

(* edited:
"31-Jul-79 20:03")

(RESETFORM ([LAMBDA (X)
(ECHOCONTROL CODE X)
(QUOTE REAL))
(PRIN1 (CHARACTER CODE))

[17]

```

(UNCRUNCH
  [LAMBDA (NUM)
    (PROG ((PTR SCRATCHFIVE))
      (* CONVERTS A SINGLE
        WORD TO A LIST OF FIVE
        CHAR CODES)
      (* Reuses a scratch list
        for efficiency)
      [RPTQ 5 (PROGN (RPLACA PTR (LRSH NUM 29))
        (SETN NUM (LLSH NUM 7))
        (SETQ PTR (CDR PTR)
      (RETURN SCRATCHFIVE])

```

[18]

```

(CRUNCH
  [LAMBDA (X)
    (PROG ((NUM 0))
      (* CONVERTS A LIST OF
        FIVE CHAR CODES TO A
        SINGLE WORD)

      (* If the list is less than 5 chars the extra
        positions are filled with blanks
        (ASCII 32))

      [RPTQ 5 (PROGN (SETN NUM (LOGOR (LLSH NUM 7)
        (OR (CAR X)
          32)))
        (SETQ X (CDR X)
      (RETURN (LLSH NUM 1))

```

[19]

```

(DSPCNVRT
  [LAMBDA (X)

    (* Converts an atom or string of up to 10 characters
    into 2 integers corresponding to the FORTRAN
    representation of the chars.
    Pads right with blanks. Returns values bound to
    DSPWORD1 and DSPWORD2.)

    (SETQ X (DCHCON X SCRATCHTEN))
    (SETQ DSPWORD2 (CRUNCH (NTH X 6)))
    (SETQ DSPWORD1 (CRUNCH X))

```

[20]

```

(DSPNUMAT
  [LAMBDA (X)
    (COND
      ((LISTP X)
        (MAPCAR X (FUNCTION DSPNUMAT)))

```

(T (PACKC (UNCRUNCH X))

[21]

(FKVALAT
[LAMBDA (ID BIAS NVALS)
(DSPNUMAT (APPLY (FUNCTION FKVALI)
(LIST ID BIAS NVALS))

[22]

(DSPADDTRH
[LAMBDA (NAME ID TYPE) (* NOBIND
"12-Dec-78 17:30")
(DSPCNVRT NAME) (* DSPCNVRT returns
output bound to DSPWORD1
and DSPWORD2)
(FKCALL DSPTRH SUBR DSPWORD1 DSPWORD2 (DSPCNVRT ID)
(DSPCNVRT TYPE])

[23]

(DSPADDINC
[LAMBDA (NAME LAT LON TIME)
(DSPCNVRT NAME)
(FKCALL DSPINC SUBR DSPWORD1 DSPWORD2 LAT LON TIME])

[24]

(DSPGRAB
[LAMBDA (TTYNO)
(NEQ 1 (COND
(TTYNO (SETQ DSPTTYCODE (IPLUS 400000Q (OCTSAMEDIGITS TTYNO))
(FKJSYS 70Q DSPTTYCODE))
(T (SETQ DSPTTYCODE 777777Q])

[25]

(DSPRELD
[LAMBDA NIL
(FKJSYS 57 DSPTTYCODE))

[26]

(DSPTTY
[LAMBDA NIL
(FKJSYS 135 (CAR FORKDATA)
(LOGOR (LLSH DSPTTYCODE 18)
DSPTTYCODE))
(FKCALL OLDMOD SUBR])

[27]

(DSPQUIET
[LAMBDA NIL
(FKJSYS 135 (CAR FORKDATA)

```

      (LOGOR (LLSH DSPTTYCODE 18)
              131071))
(FKCALL DSPMOD SUBR])

```

[28]

```

(BKDSPBUF
 [LAMBDA (X)
  (MAPC (CHCON X)
    (FUNCTION (LAMBDA (C)
      (FKJSYS 76 DSPTTYCODE (COND
        ((EQ C 31)
          13)
        (T C))

```

[29]

```

(DECSAMEDIGITS
 [LAMBDA (X)

```

```

(* Converts an octal
number to a decimal with
the same digits)

```

```

(COND
  ((LESSP X 8)
   X)
  (T (IPLUS (IREMAINDER X 8)
    (ITIMES 10 (DECSAMEDIGITS (IQUOTIENT X 8))

```

[30]

```

(OCTSAMEDIGITS
 [LAMBDA (X)

```

```

(* Converts a decimal number, all of whose digits
are less than eight, to an octal number having the
same digits)

```

```

(COND
  ((LESSP X 12Q)
   X)
  (T (IPLUS (IREMAINDER X 12Q)
    (ITIMES 10Q (OCTSAMEDIGITS (IQUOTIENT X 12Q))

```

[31]

```

(DSPTTYSTR
 [LAMBDA NIL
  (CONCAT "TTY" [COND
    ((EQP DSPTTYCODE 262143)
     "")
    (T (DECSAMEDIGITS (IDIFFERENCE DSPTTYCODE 131072)
      ":", "])

```

[32]

```
(TEKCOM
[LAMBDA (STR)

(PRIN1 TEKCOMCHAR)
(PRIN1 STR)
(TERPRI))
```

(* edited:
"30-Jul-79 19:06")

[33]

```
(TEKTEST
[LAMBDA NIL
```

(* edited:
"10-Aug-79 16:19")

```
(PROG (UTEKFLG)
(CLEARBUF)
(PRINCHAR 27)
(PRINCHAR 5)
(TERPRI)
(DISMISS 2000)
(SETQ TEKFLG (READP T))
(CLEARBUF)
(PRIN1 "Are you running on a Tektronix?")
(SETQ UTEKFLG (EQ (ASKUSER)
(QUOTE Y)))
(SETQ TEK4025 NIL)
[COND
((AND UTEKFLG (NOT TEKFLG))
(PRIN1 "TEK4025? ")
(SETQ TEK4025 (EQ (ASKUSER)
(QUOTE Y)))
(COND
(TEK4025 (PRIN1
"Please type TEK4025 command character: ")
(SETQ TEKCOMCHAR (READ))
(CLEARBUF)
(RETURN (SETQ TEKFLG UTEKFLG))
```

[34]

```
(TEKWAIT
[LAMBDA NIL
```

(* edited:
" 9-Feb-79 16:51")

```
(COND
(TEKFLG (JSYS 68 DSPTTYCODE))
```

[35]

```
(DSPERASE
[LAMBDA NIL
```

(* NOBIND
"18-Dec-78 17:32")

```
(FKCALL ERASE SUBR)
(TEKWAIT)
```

[36]

```
(DSPEXCH
[LAMBDA (NAME)
  (RPTQ 10 (RPLSTRING DSPEXCHBUF (ADD1 (ITIMES RPTN 5))
           (OR (NTHCHAR NAME RPTN)
                " ")))
(FKCALL DSPLAX SUBR DSPEXCHBUF])
```

[37]

```
(DSPMAP
[LAMBDA NIL
(COND
  ((INFILE (QUOTE HGHRES.MER))
   (FKSETVAL FLAGS 3 (DSPCNVRT "MAP"))))
(T (PRIN1 "File missing: HGHRES.MER -- Map not available")
  (TERPRI))
```

[38]

```
(DSPNOMAP
[LAMBDA NIL
(FKSETVAL FLAGS 3 (DSPCNVRT "NOMAP"))]
```

[39]

```
(DSPTOP
[LAMBDA (WAITFLG)
  (PROG ((DSPNOWAITFLG T)
         FIRSTCMD)
    [COND
      ((OR WAITFLG (EQP DSPTTYCODE 262143))
       (SETQ DSPNOWAITFLG (GETTOPVAL (QUOTE DSPNOWAITFLG)
                                       (RESETLST [COND
                                                  (TEK4025 (RESETSAVE (GRATEK)
                                                                           (QUOTE (MONTEK)
                                                                           (FKCALL DSPTOP SUBR]))
```

[40]

```
(DSPSTAT
[LAMBDA NIL
(FKJSYS 156Q (CAR FORKDATA))
(LRSH FKJSYSAC1 22Q)]
```

[41]

```
(DSPADDINCS
[LAMBDA (NAME INCLST)
  (DSPCNVRT NAME)
  (MAPC INCLST (FUNCTION (LAMBDA (INC)
                           (FKCALL DSPINC SUBR DSPWORD1 DSPWORD2 (CAR INC)
                               (CADR INC)
                               (CADDR INC))
```

[42]

```

(DSPCHGTRH
  [LAMBDA (NAME ID TYPE)
    (DSPQUIET)
    (DSPEXCH NAME)
    (AND ID (FKSETVAL CLOC 26 (DSPCNVRT ID)))
    (AND TYPE (FKSETVAL CLOC 27 (DSPCNVRT TYPE)))
    (FKCALL DSPLAX SUBR DSPEXCHEMP)
    (DSPTTY)]

```

[43]

```

(DSPSAVE
  [LAMBDA NIL
    (FKCALL DREL SUBR)
    (DSPRELD)
    (PROG1 (FKSAVE (QUOTE DSPLIB.EXE))
      (FKKILL))
  )
(DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVAR
(ADDTOVAR NLAMA M)
(ADDTOVAR NLAML FKVALAT)
(ADDTOVAR LAMA )
)
(RPAQQ SCRATCHTEN (0 0 0 0 0 0 0 0 0 0))
(RPAQQ SCRATCHFIVE (0 0 0 0 0))
(RPAQ DSPEXCHBUF (CONCAT
  "
  (CHARACTER 0)))
(RPAQ DSPEXCHEMP (CONCAT
  "      E      M      P      T      Y
  (CHARACTER 0)))
(DECLARE: DONTCOPY
  (FILEMAP (NIL (1012 9856 (GRATEK 1024 . 1312) (M 1316 . 1404) (MONTEK
1408 . 1526) (DSPCMD 1530 . 2019) (DSPINIT 2023 . 2635) (PRINCHAR 2639 .
2843) (UNCRUNCH 2847 . 3303) (CRUNCH 3307 . 3848) (DSPCNVRT 3852 . 4357
) (DSPNUMAT 4361 . 4479) (FKVALAT 4483 . 4589) (DSPADDTRH 4593 . 4930) (
DSPADDINC 4934 . 5081) (DSPGRAB 5085 . 5310) (DSPRELD 5314 . 5381) (
DSPTTY 5385 . 5519) (DSPQUIET 5523 . 5655) (BKDSPBUF 5659 . 5838) (
DECSAMEDIGITS 5842 . 6099) (OCTSAMEDIGITS 6103 . 6396) (DSPTTYSTR 6400 .
6609) (TEKCOM 6613 . 6763) (TEKTEST 6767 . 7771) (TEKWAIT 7775 . 7933)
(DSPERASE 7937 . 8073) (DSPEXCH 8077 . 8266) (DSPMAP 8270 . 8504) (
DSPNOMAP 8508 . 8575) (DSPTOP 8579 . 9047) (DSPSTAT 9051 . 9162) (
DSPADDINCS 9166 . 9423) (DSPCHGTRH 9427 . 9649) (DSPSAVE 9653 . 9853))))
)
STOP

```

(FILECREATED "18-Dec-78 16:30:43" <PMORRIS>FORK.LSP.19 40382

changes to: FORKCOMS

previous date: "29-Nov-78 17:47:22" <PMORRIS>FORK.LSP.18)

(PRETTYCOMPRINT FORKCOMS)

```

(RPAQQ FORKCOMS [(VARS (DSPNOWAITFLG NIL))
  (FNS * FORKFNS)
  [ADDVARS (GLOBALVARS FORKDATA DSPNOWAITFLG)
    (AFTERSYSOUTFORMS (PROGN (RPLACA FORKDATA NIL)
      (FKKILL]
  (P (AND (EQ (EVALV (QUOTE FORKDATA))
    (QUOTE NOBIND))
    (SETQ FORKDATA NIL)))
  (PROP MACRO FKIDPB FKRACS FKSACS PUTTYP FKHANDL FKHT FKSHR
    FKSVMACS FKDDT FKJFN FKHT_ FKDDT_ FKPROG FKHALT)
  (BLOCKS * FORKBLOCKS)
  (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS
    (ADDVARS (NLAMA SAILCALL FKCALL)
      (NLAML FKSETVAL FKVALI FKVALR FKVAL FKSETA
        FKELTR FKELTI FKELT FKARRAY FKX
        FKINIT]))

```

(RPAQ DSPNOWAITFLG NIL)

```

(RPAQQ FORKFNS (FKINIT FKKILL FKSAVE FKDDT FKCALL FKCATYPE FKSR
  SAILCALL SAILARG SAILSTRING FKACS FKACSRETURN
  FKRTN NOFORK FKCALLERR FKSX FKX FKTTYSET FKARRAY
  FKCORGET FKELT FKELTI FKELTR FKSETA FKARRAYP
  FKARRAYSIZE SAILARRAYSIZE FKARRAYTYPE FKBCHECK
  FKARRADR FKFLOAT ARRLOC FKVAL FKVALR FKVALI
  FKSETVAL FKSVM FKSVMPT FKSVMPT GETRADIX50 FKTIME
  FKJSYS FKJSYSARG FKWAIT))

```

(DEFINEQ

[44]

```

(FKINIT
  [NLAMBDA (PROGRAM)
    (PROG ((FKACS (ARRAY 20Q 20Q))
      (FKHT (HARRAY 62Q))
      (FKSVMACS (ARRAY 20Q 20Q))
      FKSHR FKJFN FKDDT FKHANDL PGS PROGFIL HALTADR HALTED EV)
    (SETQ PROGFIL
      (OR (INFILEP PROGRAM)
        (AND (NOT (MEMBER (QUOTE %)
          (UNPACK PROGRAM)))
          (INFILEP (PACK (LIST PROGRAM (QUOTE .EXE)
            (ERROR PROGRAM " FILE NOT FOUND"))))
        (AND (EQP (FKJSYS 20Q 100001000000Q (MKSTRING PROGFIL))
          1)
          (ERROR PROGRAM " -- GTJFN FAILED"))
      (SETQ FKJFN FKJSYSAC1)

```

```

(COND
  ((EQP (FKJSYS 152Q 200000000000Q)
    1)
    (* FAILED TO CREATE
    FORKNAME)
    (* RELEASE FKJFN)
    (FKJSYS 23Q FKJFN)
    (ERROR PROGRAM " -- CANNOT CREATE FORK"))
  (SETQ FKHNDL FKJSYSAC1)
  (FKJSYS 200Q (LOGOR (LLSH FKHNDL 22Q)
    FKJFN))
    (* GET)
    (* GEVEC)
  (FKJSYS 205Q FKHNDL)
  (SETQ HALTED (LRSH FKJSYSAC2 22Q))
    (* HALTED IS A TEMPORARY
    VARIABLE HERE)

[COND
  ((OR (ILESSP HALTED 5)
    (IGREATERP HALTED 17Q))
    (FKSW FKHNDL 124Q)
    (* REENTER)
    (FKJSYS 205Q FKHNDL)
    (SETQ HALTED (LRSH FKJSYSAC2 22Q))
    (* GEVEC AGAIN)

  (COND
    ((OR (ILESSP HALTED 5)
      (IGREATERP HALTED 17Q))
      (ERROR PROGRAM " -- UNABLE TO INITIALIZE LOWER FORK")
    (SETQ EV FKJSYSAC2)
    (SETQ HALTED (RESETFORM (FKTTYSET (QUOTE INITIAL))
      (FKSW FKHNDL 3)))
      (* START THE FORK)

  (FKJSYS 204Q FKHNDL EV)
  (FKRACS FKHNDL FKACS)
  (SETQ HALTADR (ADD1 (ELT FKACS 3)))
  (OR (EQP HALTED HALTADR)
    (ERROR PROGRAM " -- INITIALIZATION UNSUCCESSFUL"))
  (SETQ PGS (ELT FKACS 2))
    (* NUMBER OF PAGES TO
    SHARE)

  (AND (IGREATERP PGS 144Q)
    (ERROR PROGRAM " ATTEMPT TO SHARE TOO MANY PAGES"))
  [AND (IGREATERP PGS 0)
    (PROG ((LISPBLOCK (LOC (GETELK PGS)))
      (SIZE (LLSH PGS 11Q))
      SOURCE DEST)
      [SETQ SOURCE (SUB1 (LOGOR -400000000000Q
        (LRSH LISPBLOCK 11Q)
      [SETQ DEST (SUB1 (LOGOR (LLSH FKHNDL 22Q)
        (ELT FKACS 1)
      (RPTQ PGS (FKJSYS 56Q (IPLUS SOURCE RPTN)
        (IPLUS DEST RPTN)
        160000000000Q))
      (RETURN (SETQ FKSHR
        (LIST SIZE
          (IDIFFERENCE (LLSH (ELT FKACS 1)
            11Q)
            LISPBLOCK)
          (IPLUS LISPBLOCK SIZE)
          LISPBLOCK]
      (SETQ FORKDATA (LIST FKHNDL (LIST FKHT HALTADR)
        (LIST FKACS)
        FKSHR FKSVMACS FKDDT FKJFN PROFILE))

```

(* FORKDATA IS SET TO A LIST OF THE FOLLOWING -
 FKHNDL : The fork handle -
 FKHT : Hash table of fork symbols and names of
 shared arrays (values are fork addresses) -
 FKHALT : Expected PC of fork termination -
 FKJFN : JFN of program in the fork -
 FKACS : List of arrays used to hold acs for fork -
 FKSHR : List containing information about shared
 pages, format is (words-left conversion ending
 start) where words-left is the number of unallocated
 words remaining in the shared pages, conversion is
 the factor to be added to lisp address to get the
 fork address of the first word of the array, ending
 is the lisp address of the word after the block of
 shared pages, and start is the lisp address of the
 first word of the block -
 FKSYMACS : Another AC array for use by FKSYPGET -
 FKDDT : The FKJFN for DDT in the fork.)

(RETURN PROGFILE)

[45]

```
(FKKILL
  [LAMBDA NIL
    (PROG (DDT SHR FKPROG)
      (SELECTQ (EVALV (QUOTE FORKDATA))
        (NIL (RETURN NIL))
        (NOBIND (RETURN (SETQ FORKDATA NIL)))
        NIL)
      (COND
        [(FKHNDL FORKDATA)
          (FKJSYS 156Q (FKHNDL FORKDATA)) (* RFSTS)
          (COND
            ((EQP (RSH FKJSYSAC1 22Q)
              -1)
              (SETQ FKPROG NIL))
            (T (SETQ FKPROG (FKPROG FORKDATA))
              (FKJSYS 153Q (FKHNDL FORKDATA))
              (* KFORK)
              (FKJSYS 23Q (FKJFN FORKDATA))
              (* RLJFN)
              (SETQ DDT (FKDDT FORKDATA))
              (AND DDT (FKJSYS 23Q DDT)) (* RLJFN)
              ]
            (T (SETQ FKPROG NIL)))
          (SETQ SHR (CDDR (FKSHR FORKDATA)))
          (AND SHR (RELBLK (VAG (CADR SHR))
            (LRSH (IDIFFERENCE (CAR SHR)
              (CADR SHR))
              11Q)))
          (SETQ FORKDATA NIL)
          (RETURN FKPROG])
```

[46]

```

(FKSAVE
  [LAMBDA (FILE)
    (* SAVE A FORK FILE ON
    THE DISK)
    (PROG (JFN)
      (OR (EQ 2 (FKJSYS 20Q -377777000000Q (MKSTRING FILE)))
        (ERROR FILE " -- GTJFN FAILED"))
      (SETQ JFN FKJSYSAC1)
      (FKJSYS 202Q (LOGOR (LLSH (FKHNDL FORKDATA)
        22Q)
        JFN)
        -17777760Q)
      (RETURN (INFILEP FILE))

```

[47]

```

(FKDDT
  [LAMBDA (DDTFILE)
    (PROG (FKHNDL)
      (OR FORKDATA (NOFORK))
      (SETQ FKHNDL (FKHNDL FORKDATA))
      [OR
        (FKDDT FORKDATA
          (* THE MACRO FKDDT))
        (PROGN
          (FKJSYS 47 (LOGOR (LLSH FKHNDL 18)
            504))
          (* CHECK IF DDT WAS
          SAVED WITH THE FORK)
          (AND (ZEROP (LOGAND FKJSYSAC2 1073741824))
            (PROG (DDT EV)
              (FKJSYS 133 FKHNDL)
              (* SAVE ENTRY VECTOR
              WORD BEFORE DO GET FOR
              DDT)
              (SETQ EV FKJSYSAC2)
              (AND (EQP (FKJSYS 16 8590196736
                (COND
                  (DDTFILE (MKSTRING DDTFILE)
                    )
                  (T "<SUBSYS>IDDT.EXE"))
                1)
                (ERROR " " "CANNOT GET JFN FOR DDT"))
              (SETQ DDT FKJSYSAC1)
              (FKJSYS 128 (LOGOR (LLSH FKHNDL 18)
                DDT))
              (FKJSYS 132 FKHNDL EV)
              (* RESTORE OLD ENTRY
              VECTOR)
              (FKDDT_ FORKDATA DDT)
              (RETURN (FKSW FKHNDL 6))
              (* SET UP $I-1 IN THE
              FORK)
            ]
          (RESETFORM (FKTTYSET T)
            (FKSW FKHNDL 258048))
            (* START FORK AT DDT
            ENTRY WAIT FOR FORK TO
            HALT)
          (RETURN T1)

```

[48]

```

(FKCALL
  [NLAMBDA FKCX
    (* NOBIND
      "24-Nov-78 13:35")
    (PROG (FKCBP FKARG FKCTYPE FKCVL FKWRDS FKCBP FKHNDL FKHT FKCA
      FKRESLIST (FKCID (CAR FKCX))
      FKBIAS FKRESULT (FKRESULTYPE (CADR FKCX))
      (FKCARGS (CDDR FKCX))
      (FKCN 3))
    (OR FORKDATA (NOFORK))
    (SETQ FKHNDL (FKHNDL FORKDATA))
    (SETQ FKHT (FKHT FORKDATA))
    (SETQ FKCA (FKACS))
    (SETQ FKCBP (LOGOR -30014439424 (IPLUS (LOC FKCA)
      2)))
    (* 3 BIT BYTE POINTER TO
      TYPE BITS)
    (SETQ FKCBP (LOGOR -29460791296 (IPLUS (LOC FKCA)
      3)))
    (* FULL WORD BYTE
      POINTER TO FKCA+3)

    (FKWAIT FKHNDL)
    (AND (LISTP FKCID)
      (SETQ FKCID (EVAL FKCID)))
    (FKIDPB (FKSYM FKCID FKHT)
      FKCBP
    (* STORE THE ADDRESS OF
      THE SUBPROGRAM IN FKCA
      (2))

    )
  ARGLOOP
    (OR FKARGS (GO ARGDONE))
    (SETQ FKARG (CAR FKARGS))
    (AND (LISTP FKARG)
      (SELECTQ (CAR FKARG)
        (BIAS (SETQ FKBIAS (EVAL (CADR FKARG)))
          (SETQ FKCTYPE (FKCATYPE FKBIAS))
          [SETQ FKARG
            (IPLUS (FKSYM FKBIAS FKHT)
              (SUB1 (EVAL (CADDR FKARG))
                (GO PUTARG))
            ((INTEGER REAL LOGICAL)
              (SETQ FKRESLIST (CONS (CONS FKCN FKARG)
                FKRESLIST))
              (SETQ FKARG (CADR FKARG)))
            NIL))
      (SETQ FKARG (EVAL FKARG))
    (COND
      ((EQ FKARG T)
        (SETQ FKARG -1)
        (SETQ FKCTYPE 3))
      ((NULL FKARG)
        (SETQ FKARG 0)
        (SETQ FKCTYPE 3))
      ((LITATOM FKARG)
        (SETQ FKCTYPE (FKCATYPE FKARG))

```

```

      (SETQ FKCARG (FKSYM FKCARG FKHT))
    ((FIXP FKCARG)
     (SETQ FKCTYPE 0))
    ((FLOATP FKCARG)
     (SETQ FKCTYPE 2))
    ((STRINGP FKCARG)
     (SETQ FKCWRDS (FKSR FKCA FKN FKCARG))
     (OR FKCWRDS (FKCALLERR FKCID))
     (GO MWARG))
    ((ARRAYP FKCARG)
     (SETQ FKCWRDS (ARRAYSIZE FKCARG))
     (AND (IGREATERP FKCWRDS (IDIFFERENCE 17 FKN))
          (FKCALLERR FKCID))
     (RPTQ FKCWRDS (SETA FKCA (IPLUS FKN RPTN -1)
                          (ELT FKCARG RPTN)))
     (GO MWARG))
    ((LISTP FKCARG)
     (ERROR FKCARG
      "LISTS CANNOT BE USED AS ARGUMENTS FOR FORK CALLS"))
    [(FKARRAYP FKCARG)
     (SETQ FKCTYPE (COND
                     ((EQ (FKARRAYTYPE FKCARG)
                          (QUOTE REAL))
                      4)
                     (T 1)))
     (SETQ FKCARG (IPLUS (LOC FKCARG)
                          (CADR (FKSHR FORKDATA)
                               (T (ERROR FKCARG " ILLEGAL ARG TYPE FOR FORK CALL"))))
    PUTARG
      (AND (IGREATERP FKN 16)
           (FKCALLERR FKCID))
      (FKIDPB FKCARG FKCABP
              (* STORE VALUE OF FKCARG
                INTO FKCA))
      (SETQ FKN (ADD1 FKN))
      (FKIDPB FKCTYPE FKCBP
              (* PUT TYPE))
    ENDARG
      (SETQ FKCARGS (CDR FKCARGS))
      (GO ARGLOOP)
    MWARG
      (SETQ FKN (IPLUS FKN FKCWRDS))
      (PUTTYP 0)
      (SETQ FKCABP (IPLUS FKCABP FKCWRDS))
      (RPTQ (SUB1 FKCWRDS)
            (PUTTYP 5))
      (GO ENDARG)
    ARGDCNE
      (PUTTYP 6)
      (FKSACS FKHNDL FKCA)
      (FKSW FKHNDL 5 T)
      (AND DSPNOWAITFLG (RETURN))
      (AND (LISTP FKRESULTYPE)
           (SETQ FKRESULTYPE (EVAL FKRESULTYPE)))
      (AND (EQ FKRESULTYPE (QUOTE SUBR))
           (NULL FKCRSLIST)
           (PROGN (FKACSRETURN FKCA)
                  (RETURN NIL)))
      (FKRACS FKHNDL FKCA)

```

```

[MAPC FKCRESLIST (FUNCTION (LAMBDA (X)
  (SET (CADDR X)
    (FKRTN (CADR X)
      FKCA
      (CAR X)
    (SETQ FKRESULT (FKRTN FKRESULTTYPE FKCA 1))
    (FKACSRETURN FKCA)
    (RETURN FKRESULT))

```

[49]

```

(FKCATYPE
[LAMBDA (FKID)
  (PROG ((C (CHCON1 FKID)))

```

```

(* IF FIRST CHARACTER OF PNAME OF FKID IS IN I TO N,
THEN TYPE IS POINTER TO INTEGER, ELSE POINTER TO
REAL)

```

```

(RETURN (COND
  (IOR (ILESSP C (CHCON1 (QUOTE I)))
    (IGREATERP C (CHCON1 (QUOTE N)
      4)
    (T 1))

```

[50]

```

(FKSR
[LAMBDA (A I STR)
  (PROG [WDS (DESTPTR (LOGOR 700000000Q (IPLUS (LOC A)
    I)))
    (SIZE (NCHARS STR))
    (ROOM (IDIFFERENCE (ARRAYSIZE A)
      (SUB1 I)

```

```

(* A IS ARRAY POINTER. I IS INDEX WHERE START
PUTTING STRING, STR. IF STR IS TOO LARGE THEN RETURN
NIL, ELSE RETURN NUMBER OF WORDS USED.
PUTS ZERO WORD AFTER THE STRING)

```

```

(SETQ WDS (ADD1 (IQUOTIENT (IPLUS SIZE 4)
  5)))
(AND (IGREATERP WDS ROOM)
  (RETURN)) (* DOESN'T FIT)
(RPTQ WDS (SETA A (IPLUS I RPTN -1)
  0))
(FKJSYS 53Q DESTPTR STR (IMINUS SIZE)) (* SOUT)
(RETURN WDS))

```

[51]

```

(SAILCALL
[INLAMBDA FKCX
  (PROG (FKCBP FKCBP FKHN DL FKHT FKCA FKCRESLIST FKRESULT

```

```

      FKRESULTBITS FKCARG FKTYPE (FKCID (CAR FKCX))
      (FKRESULTTYPE (CADR FKCX))
      (FKCARGS (CDDR FKCX))
      (FKCN 4))
[SETQ FKHNDL (FKHNDL (OR FORKDATA (NOFORK)
      (SETQ FKHT (FKHT FORKDATA))
      (SETQ FKCA (FKACS))
      (SETQ FKCBP (LOGOR -337400000000Q (IPLUS (LOC FKCA)
      3)))
      (* 4 BIT BYTE POINTER TO
      FKTYPE BITS)
      (SETQ FKCBP (LOGOR -333400000000Q (IPLUS (LOC FKCA)
      5)))
      (* FULL WORD BYTE
      POINTER TO ARGUMENT
      LIST)
(COND
  [(FMEMB FKRESULTTYPE (QUOTE (SUBR REAL INTEGER BOOLEAN
      LOGICAL]
      ((EQ FKRESULTTYPE (QUOTE STRING))
      (ERROR FKRESULTTYPE
      "STRING PROCEDURES NOT IMPLEMENTED YET"))
      (T (ERROR FKRESULTTYPE "ILLEGAL TYPE FOR SAIL CALL"))
      (SETQ FKRESULTBITS (COND
      (FKTTYSETCALLED -10000000000Q)
      (T 0)))
      (SETA FKCA 1 (LOGOR FKRESULTBITS (FKSYM FKCID FKHT)))
ARGLOOP
      (OR FKCARGS (GO ARGDONE))
      (SETQ FKARG (SAILARG (CAR FKCARGS)
      FKHT))
      (SETQ FKTYPE (CAR FKARG))
      (AND (CADDR FKARG)
      (SETQ FKCRESLIST (CONS (CONS FKCN (CADDR FKARG))
      FKCRESLIST)))
      (SETQ FKARG (CADR FKARG))
      [COND
      ((ATOM FKARG)
      (AND (IGREATERP FKCN 20Q)
      (FKCALLERR FKCID))
      (FKIDPB FKARG FKCBP)
      (SETQ FKCN (ADD1 FKCN))
      (FKIDPB FKTYPE FKCBP)
      (T (AND (IGREATERP (SETQ FKCN (IPLUS FKCN (LENGTH FKARG)))
      21Q)
      (FKCALLERR FKCID))
      (MAPC FKARG (FUNCTION (LAMBDA (WORD)
      (FKIDPB WORD FKCBP)
      (FKIDPB FKTYPE FKCBP)
      (SETQ FKTYPE 14Q]
      (SETQ FKCARGS (CDR FKCARGS))
      (GO ARGLOOP)
ARGDONE
      (PUTTYP 15Q)
      (FKSACS FKHNDL FKCA)
      (FKSW FKHNDL 7)
      (AND (EQ FKRESULTTYPE (QUOTE SUBR))
      (* END OF ARGS)
      (* CALL THE FUNCTION)

```

```

      (NULL FKRESLIST)
      (PROGN (FKACSRETURN FKCA)
              (RETURN NIL)))
(FKRACS FKHNDL FKCA)
[MAPC FKRESLIST (FUNCTION (LAMBDA (X)
      (SET (CADDR X)
            (FKRTN (CADR X)
                    FKCA
                    (CAR X)
      (SETQ FKRESULT (FKRTN FKRESULTTYPE FKCA 1))
      (FKACSRETURN FKCA)
      (RETURN FKRESULT))

```

[52]

```

(SAILARG
  [LAMBDA (FKARG FKHT)

```

```

    (* FKTYPE BITS -- 1 : STRING, 2 : REFERENCE
    (TO PROCEDURE), 4 : REFERENCE
    (TO LOWER FORK), 10Q : FKARRY)

```

```

(PROG ((FKRV (QUOTE VALUE))
      (FKTYPE 0)
      FKARRY VARTYPE FKVALUE FKRESULTS FKVARBL FKCALLTYPE)
[COND
  [[AND (LISTP FKARG)
        (ATOM (CAR FKARG))
        (FMEMB (CAR FKARG)
                (QUOTE (REFERENCE VALUE INTEGER REAL BOOLEAN
                        LOGICAL STRING ARRAY)
      (SETQ FKVARBL (CAR (LAST FKARG))
      (T (SETQ FKVARBL FKARG)
        (SETQ FKARG (LIST FKARG))
      (SETQ FKVALUE (EVAL FKVARBL))
      (* INSPECT THE ARGUMENT)
      (COND
        ((EQ FKVALUE T)
          (SETQ FKVALUE -1)
          (SETQQ VARTYPE INTEGER))
        ((EQ FKVALUE NIL)
          (SETQ FKVALUE 0)
          (SETQQ VARTYPE INTEGER))
        ((LITATOM FKVALUE)
          (SETQ FKVALUE (FKSYM FKVALUE FKHT))
          (SETQ FKTYPE 4))
        ((STRINGP FKVALUE)
          (SETQ FKVALUE (SAILSTRING FKVALUE))
          (SETQQ VARTYPE STRING))
        ((FKARRAYP FKVALUE)
          [SETQ FKVALUE (IPLUS (LOC FKVALUE)
                                (CADR (FKSHR FORKDATA)
          (SETQ FKTYPE 12Q))
        ((FIXP FKVALUE)
          (SETQQ VARTYPE INTEGER))
        ((FLOATP FKVALUE)
          (SETQQ VARTYPE REAL))

```

```

((LISTP FKVALUE)
 (ERROR FKVALUE
  "LISTS CANNOT BE USED AS ARGUMENTS FOR FORK CALLS"))
(T (ERROR FKVALUE "ILLEGAL ARG TYPE FOR SAIL CALL"))
    (* INSPECT THE
    MODIFIERS)
(SETQ FKCALLTYPE (OR VARTYPE (QUOTE INTEGER)))
[MAP FKARG (FUNCTION (LAMBDA (X)
  (AND (CDR X)
    (SELECTQ (CAR X)
      (REFERENCE (SETQ FKR V REFERENCE))
      (VALUE (SETQ FKR V VALUE))
      (INTEGER (SETQ FKCALLTYPE INTEGER))
      (REAL (SETQ FKCALLTYPE REAL))
      ((BOOLEAN LOGICAL)
       (SETQ FKCALLTYPE LOGICAL))
      (STRING (SETQ FKCALLTYPE STRING))
      (ARRAY (SETQ FKARRY T))
      (ERROR (CAR X))

      "ILLEGAL ARGUMENT TYPE FOR SAIL CALL"]
    (* PERFORM FKTYPE
    CONVERSIONS)
(AND VARTYPE (NEQ FKCALLTYPE VARTYPE)
 (PROGN (COND
  ((EQ VARTYPE (QUOTE STRING))
   (SETQ FKVALUE (LLSH (CAR FKVALUE)
    -35Q))
   (SETQ VARTYPE INTEGER)))
 (COND
  ((AND (EQ VARTYPE (QUOTE REAL))
        (EQ FKCALLTYPE (QUOTE INTEGER)))
   (SETQ FKVALUE (FIX FKVALUE)))
  ((AND (EQ VARTYPE (QUOTE INTEGER))
        (EQ FKCALLTYPE (QUOTE REAL)))
   (SETQ FKVALUE (FLOAT FKVALUE)))
  ((EQ FKCALLTYPE (QUOTE STRING))
   (AND (EQ VARTYPE (QUOTE REAL))
        (SETQ FKVALUE (FIX FKVALUE)))
   (SETQ FKVALUE (LLSH FKVALUE 35Q)
    (* SET FKTYPE BITS)
(AND (EQ FKCALLTYPE (QUOTE STRING))
 (SETQ FKTYPE (LOGOR FKTYPE 1)))
(AND (EQ FKR V (QUOTE REFERENCE))
 (SETQ FKTYPE (LOGOR FKTYPE 2))
 (LITATOM FKVARBL)
 (SETQ FKRESULTS (LIST FKCALLTYPE FKVARBL)))
(AND FKARRY (SETQ FKTYPE (LOGOR 12Q FKTYPE)))
(RETURN (LIST FKTYPE FKVALUE FKRESULTS])

```

[53]

```

(SAILSTRING
 [LAMBDA (STRING)

```

```

(* THIS COULD PROBABLY
BE DONE MUCH MORE
QUICKLY)

```

```

(PROG (VAL ZEROS CHLIST PACKEDLIST)

```

```

      (SETQ CHLIST (CHCON STRING))
STRINGLOOP
  [OR CHLIST (PROGN (OR ZEROS (SETQ PACKEDLIST
                              (CONS 0 PACKEDLIST)))
                  (RETURN (REVERSE PACKEDLIST)

      (SETQ VAL 0)
      [RPTQ 5 (PROGN (SETQ VAL (IPLUS (LLSH VAL 7)
                                      (OR (CAR CHLIST)
                                          0)))

                  (OR (CAR CHLIST)
                      (SETQ ZEROS T))
                  (SETQ CHLIST (CDR CHLIST)

      (SETQ PACKEDLIST (CONS (LLSH VAL 1)
                              PACKEDLIST))

      (GO STRINGLOOP])

```

[54]

```

(FKACS
  [LAMBDA NIL
    (PROG ((Y (CDDR FORKDATA))
           X)
      (RETURN (COND
                ((SETQ X (CAR Y))
                 (RPLACA Y (CDR X))
                 (CAR X))
                (T (ARRAY 20Q 20Q))

```

[55]

```

(FKACSRETURN
  [LAMBDA (ARRAY)
    (PROG ((Y (CDDR FORKDATA)))
      (RETURN (RPLACA Y (CONS ARRAY (CAR Y))

```

[56]

```

(FKRTN
  [LAMBDA (TYPE A N)
    (SELECTQ TYPE
      (INTEGER (ELT A N))
      (REAL (ASSEMBLE NIL
                  (CQ (VAG (IPLUS (LOC A)
                              N)))
                  (MOVE 1 , 1 (1))
                  (FASTCALL MKFN)))
      [LOGICAL (NOT (ZEROP (ELT A N)
                          (SUBR NIL)
                          (ERROR TYPE "ILLEGAL RESULT TYPE FOR FORK CALL"))

```

[57]

```

(NOFORK
  [LAMBDA NIL
    (PRIN1 "NO FORK!
PROGRAM NAME: ")
    (APPLY* (FUNCTION FKINIT)

```

```

      (READ))
    FORKDATA])

```

[58]

```

(FKCALLERR
 [LAMBDA (FKCID)
  (ERROR FKCID " TOO MANY WORDS OF ARGS FOR A FORK CALL")])

```

[59]

```

(FKSW
 [LAMBDA (FKHNDL I FKNOWAITFLG)
  (* NOBIND
   "24-Nov-78 12:43")

  (PROG ((EXPECTED (FKHALT FORKDATA))
         HALTED)
    (COND
      ((ILESSP I 33)
       (FKJSYS 129 FKHNDL I)
       )
      (T (FKJSYS 111 FKHNDL I)
       ))
    (AND FKNOWAITFLG DSPNOWAITFLG (RETURN EXPECTED))
    (FKJSYS 115 FKHNDL)
    (FKJSYS 110 FKHNDL)
    (SETQ HALTED (LOGAND FKJSYSAC2 262143))
    (AND EXPECTED (NULL (EQP HALTED EXPECTED))
     (RESETFORM (RADIX 8)
      (HELP "LOWER FORK HALTED AT ADDRESS: " HALTED)
      ))
    (RETURN HALTED]))

```

[60]

```

(FKX
 [NLAMBDA (FKCX)
  (EVAL (LIST (QUOTE RESETFORM)
              (QUOTE (FKTTYSET T))
              FKCX)])

```

[61]

```

(FKTTYSET
 [LAMBDA (BOOL)

  (* IF BOOL IS T, DISARMS LISP CONTROL CHARACTER
    INTERRUPTS, EXCEPT FOR ^B, ^D, ^E, AND ^H.
    IF BOOL IS NIL, RESTORES INTERRUPTS AND TERMINAL
    CHARACTERISTICS)

```

```

(COND
  ((EQ BOOL (QUOTE INITIAL))
   (FKJSYS 112Q 101Q)
   (SETQ FKCC1 FKJSYSAC2)
   (SETQ FKCC2 FKJSYSAC3)
   (FKJSYS 107Q 101Q)

```

(* RFCOC)

(* RFMOD)

```

      (SETQ FKFMOD FKJSYSAC2)
      (FKJSYS 173Q 400000Q)
      (SETQ FKTIW FKJSYSAC2)
      (FKJSYS 174Q 400000Q 131000000000Q)
      (SETQ FKTTYSETCALLED T)
      NIL)
      (BOOL (FKJSYS 174Q 400000Q 131000000000Q)
      (SETQ FKTTYSETCALLED T)
      NIL)
      (T (FKJSYS 113Q 101Q FKCC1 FKCC2)
      (FKJSYS 110Q 101Q FKFMOD)
      (FKJSYS 174Q 400000Q FKTIW)
      (SETQ FKTTYSETCALLED NIL)
      T))

```

(* RTIW)

(* STIW FOR ^B, ^D, ^E,
AND ^H)

(* STIW)

(* SFCOC)

(* SFMOD)

(* STIW)

[62]

```

(FKARRAY
  [NLAMBDA (FKA FKTYPE FKSIZE FKSIZE2)
    (PROG ((FKOFFSET 0)
      (FKTOTALSIZE 1)
      (FKNDIM 0)
      [FKSIZES (COND
        ((NLISTP FKSIZE)
          (EVAL FKSIZE))
        ((GETD (CAR FKSIZE))
          (EVAL FKSIZE))
        (T (MAPCAR FKSIZE (FUNCTION EVAL)
          FKDIMS FKDOPE FKHI FKLO FKBYTP FKDATAWD FKLOC)
          (OR FORKDATA (NOFORK))
          (SETQ FKTYPE (SELECTQ FKTYPE
            (REAL -1)
            (INTEGER 0)
            (ERROR FKA
              " HAS ILLEGAL TYPE DECLARATION"))))
        (COND
          (FKSIZE2 (SETQ FKHI (EVAL FKSIZE2))
            (SETQ FKOFFSET (ADD1 FKSIZES))
            (SETQ FKNDIM 2)
            (SETQ FKTOTALSIZE (ITIMES FKSIZES FKHI))
            (SETQ FKDOPE (LIST 1 FKSIZES 1 FKSIZES FKHI 1))
            (GO FKDIMDONE)))
          [SETQ FKDIMS (COND
            ((LISTP FKSIZES)
              (REVERSE FKSIZES))
            (T (LIST FKSIZES 1)
              FKDIMLOOP
                (OR (AND (NUMBERP (SETQ FKHI (CAR FKDIMS)))
                  (NUMBERP (SETQ FKLO (CADR FKDIMS)))
                  (NOT (ILESSP FKHI FKLO)))
                  (ERROR FKSIZES "INVALID INDEX SPECIFICATION"))
                [SETQ FKDOPE (CONS FKTOTALSIZE (CONS FKHI (CONS FKLO FKDOPE)
                  (SETQ FKOFFSET (IPLUS FKOFFSET (ITIMES FKTOTALSIZE FKLO)))
                  (SETQ FKNDIM (ADD1 FKNDIM))
                  [SETQ FKTOTALSIZE (ITIMES FKTOTALSIZE

```

```

                                (ADD1 (IDIFFERENCE FKHI FKLO)
(AND (SETQ FKDIMS (CDDR FKDIMS))
      (GO FKDIMLOOP))
FKDIMDONE
  (SETQ FKLOC (FKCORGET (IPLUS FKTOTALSIZE (ITIMES FKNDIM 3)
                                4)))
  (SETQ FKBYTP (LOGOR -333400000000Q FKLOC))
  (FKIDPB (SETQ FKDATAWD (IPLUS FKLOC (ITIMES FKNDIM 3)
                                4))
    FKBYTP)
                                (* POINTS TO FIRST DATA
                                WORD)
  (FKIDPB FKTYPE FKBYTP)
                                (* TYPE -- POINTS TO
                                LITERAL ATOM -- SHOULDNT
                                MOVE DURING GARBAGE
                                COLLECTION)

  [FKSYMPUT (FKHT FORKDATA)
    FKA
    (IPLUS FKDATAWD (CADR (FKSHR FORKDATA)
    (FKIDPB (IPLUS FKDATAWD (CADR (FKSHR FORKDATA))
      (IMINUS FKOFFSET))
    FKBYTP)
                                (* BASE ADDRESS FOR SAIL
                                ADDRESS CALCULATION)

  [MAPC (REVERSE FKDOPE)
    (FUNCTION (LAMBDA (WORD)
      (FKIDPB WORD FKBYTP)
                                (* FOR EACH DIMENSION,
                                LOWERBOUND, UPPER BOUND,
                                MULTIPLIER)

  (FKIDPB (LOGOR (LLSH FKNDIM 22Q)
    FKTOTALSIZE)
    FKBYTP)
                                (* XWD NDIMS,,TOTAL
                                FKTOTALSIZE)

  (RETURN (SET FKA (VAG FKDATAWD]))

```

[63]

```

(FKCORGET
  [LAMBDA (SIZE)
    (PROG ((SHR (FKSHR FORKDATA))
      X)
      (AND (IGREATERP SIZE (CAR SHR))
        (ERROR (FKPROG FORKDATA)
          " SHARED PAGES EXCEEDED"))
      (SETQ X (IDIFFERENCE (CADDR SHR)
        (CAR SHR)))
      (RPLACA SHR (IDIFFERENCE (CAR SHR)
        SIZE))
      (RETURN X))

```

[64]

```

(FKELT
  [NLAMBDA (FKELTIA FKELTIN FKELTIWORDS)
    (APPLY* (SELECTQ (FKARRAYTYPE (EVAL FKELTIA))
      (REAL (FUNCTION FKELTR))
      (INTEGER (FUNCTION FKELTI))
      (QUOTE ARRAYSCLOBBED!))
    FKELTIA FKELTIN FKELTIWORDS))

```

[65]

```

(FKELTI
  [NLAMBDA (FKELTI!A FKELTI!N FKELTI!WORDS)
    (PROG (PTR ANS)
      (SETQ FKELTI!WORDS (EVAL FKELTI!WORDS))
      (SETQ PTR (FKARRADR FKELTI!A FKELTI!N FKELTI!WORDS))
      (RETURN (COND
        [FKELTI!WORDS (RPTQ FKELTI!WORDS
          (SETQ ANS
            (CONS (OPENR (IPLUS PTR RPTN
              -1))
              ANS]
          (T (OPENR PTR]))

```

[66]

```

(FKELTR
  [NLAMBDA (FKELTR!A FKELTR!N FKELTR!WORDS)
    (PROG (PTR ANS)
      (SETQ FKELTR!WORDS (EVAL FKELTR!WORDS))
      (SETQ PTR (FKARRADR FKELTR!A FKELTR!N FKELTR!WORDS))
      (RETURN (COND
        [FKELTR!WORDS (RPTQ FKELTR!WORDS
          (SETQ ANS
            (CONS (FKFLOAT (IPLUS PTR
              RPTN -1))
              ANS]
          (T (FKFLOAT PTR]))

```

[67]

```

(FKSETA
  [NLAMBDA (FKARRY FKINDEX FKEXPR)
    (PROG (FKPTR FKVAL)
      (SETQ FKVAL (EVAL FKEXPR))
      [SETQ FKPTR (FKARRADR FKARRY FKINDEX (AND (LISTP FKVAL)
        (LENGTH FKVAL))
      (RETURN (COND
        [(LISTP FKVAL)
          (MAPCAR FKVAL (FUNCTION (LAMBDA (FKV)
            (PROG1 (CLOSER FKPTR FKV)
              (SETQ FKPTR (ADD1 FKPTR)
                ((CLOSER FKPTR FKVAL)
                  FKVAL)))

```

[68]

```

(FKARRAYP
  [LAMBDA (A)
    (PROG [(SHR (CDDR (FKSHR FORKDATA)
      (RETURN (AND SHR (IGREATERP (CAR SHR)
        (LOC A))
        (NOT (IGREATERP (CADR SHR)
          (LOC A)))
        A))

```

[69]

```
(FKARRAYSIZE
  [LAMBDA (A)
    (LOGAND 777777Q (OPENR (SUB1 (LOC A))
```

[70]

```
(SAILARRAYSIZE
  [LAMBDA (A)
    (PROG ((X (LOC A))
      ANS NDIM)
      (SETQ NDIM (LRSH (OPENR (SUB1 X))
        22Q))
      [RPTQ NDIM (PROGN (SETQ X (IDIFFERENCE X 3))
        (SETQ ANS
          (CONS (OPENR X)
            (CONS (OPENR (SUB1 X))
              ANS))
        (RETURN (REVERSE ANS))
```

[71]

```
(FKARRAYTYPE
  [LAMBDA (A)
    (PROG (NDIM)
      (SETQ NDIM (LRSH (OPENR (SUB1 (LOC A)))
        22Q))
      (RETURN (COND
        ([ZEROP (OPENR (IDIFFERENCE (LOC A)
          (IPLUS (ITIMES NDIM 3)
            3])
          (QUOTE INTEGER))
          (T (QUOTE REAL))
```

[72]

```
(FKBCHECK
  [LAMBDA (N LO HI)
    (AND (OR (IGREATERP N HI)
      (ILESSP N LO))
      (ERROR N " INDEX OUT OF RANGE"))
```

[73]

```
(FKARRADR
  [LAMBDA (FKARRNAME FKINDEX FKNWORDS)
    (PROG ((FKARRY (EVAL FKARRNAME))
      FKADR FKNDIM FKSIZE FKLOW (FKOFFSET 0)
      FKDIMS FKPTR)
      (OR (FKARRAYP FKARRY)
        (ERROR FKARRY " -- ARG NOT SHARED ARRAY"))
      (SETQ FKADR (LOC FKARRY))
      [SETQ FKDIMS (COND
        ((NLISTP FKINDEX)
          (EVAL FKINDEX))
```

```

      ((GETD (CAR FKINDEX))
       (EVAL FKINDEX))
      (T (MAPCAR FKINDEX (FUNCTION EVAL)
[COND
  ((NUMBERP FKDIM)
   [SETQ FKSIZE (LOGAND 777777Q (OPENR (SUB1 FKADR)
   (FKBCHECK FKDIM 1 FKSIZE)
   (SETQ FKOFFSET (SUB1 FKDIM)))
  (T (SETQ FKNDIM (LRSH (OPENR (SUB1 FKADR)
      22Q))
    (OR (EQP FKNDIM (LENGTH FKDIM))
      (ERROR FKINDEX
        " WRONG NUMBER OF DIMENSIONS FOR ARRAY"))
    (SETQ FKPTR FKADR)
    [MAPC FKDIM (FUNCTION (LAMBDA (X)
      (SETQ FKPTR (IDIFFERENCE FKPTR 3))
      (SETQ FKLOW (OPENR (SUB1 FKPTR)))
      (FKBCHECK X FKLOW (OPENR FKPTR))
      (SETQ FKOFFSET
        (IPLUS FKOFFSET (ITIMES (OPENR (ADD1 FKPTR))
          (IDIFFERENCE X FKLOW)
        (AND FKNWORDS (SETQ FKSIZE (LOGAND 777777Q
          (OPENR (SUB1 FKADR)
        (AND FKNWORDS (FKBCHECK (IPLUS FKOFFSET FKNWORDS)
          1 FKSIZE))
        (RETURN (IPLUS FKADR FKOFFSET])

```

[74]

```

(FKFLOAT
 [LAMBDA (ADR)
  (ASSEMBLE NIL
    (CQ (VAG ADR))
    (MOVE 1 , 0 (1))
    (FASTCALL MKFN])

```

[75]

```

(ARRLOC
 [LAMBDA (ARR)
  (COND
    ((ARRAYP ARR)
     (IPLUS 2 (LOC ARR)))
    ((FKARRAYP ARR)
     (LOC ARR))
    ((ERROR ARR "ARG NOT ARRAY"))

```

[76]

```

(FKVAL
 [NLAMBDA (FKADR FKBIAS FKWORDS)
  (APPLY* (FUNCTION FKVALI)
    FKADR FKBIAS FKWORDS (COND
      ((EQ (FKCATYPE FKADR)
        4)
       (QUOTE REAL))

```

[77]

```
(FKVALR
  [NLAMBDA (FKADR FKBIAS FKWORDS)
    (APPLY* (FUNCTION FKVALI)
      FKADR FKBIAS FKWORDS (QUOTE REAL))
```

[78]

```
(FKVALI
  [NLAMBDA (FKADR FKBIAS FKWORDS FKREAL)
    (PROG (FKHNDL FKHT FKACS FKBP FKRESULT)
      (OR FORKDATA (NOFORK))
      (SETQ FKHNDL (FKHNDL FORKDATA))
      (FKWAIT FKHNDL)
      (SETQ FKHT (FKHT FORKDATA))
      (SETQ FKACS (FKACS))
      (SETQ FKBP (LOGOR -29460791296 (IPLUS (LOC FKACS)
        2)))
      (* NOBIND
        "24-Nov-78 12:58")
      (* FULL WORD POINTER TO
        FIRST WORD OF FKACS)
      (* ONE ARGUMENT, POINTER
        TYPE)

      (FKIDPB 15032385536 FKBP)

      (FKIDPB (FKSYM (QUOTE FKVAL)
        FKHT)
        FKBP)
      (FKIDPB (IPLUS -1 (OR (EVAL FKBIAS)
        1)
        (FKSYM FKADR FKHT))
        FKBP)
      (* ADDRESS OF COMMON OR
        VARIABLE)

      (FKSACS FKHNDL FKACS)
      (FKSW FKHNDL 5)
      (FKRACS FKHNDL FKACS)
      (SETQ FKBP (IPLUS (LOC FKACS)
        4))

      [COND
        [FKWORDS (SETQ FKWORDS (EVAL FKWORDS))
          (COND
            ((IGREATERP FKWORDS 14)
              (HELP FKWORDS
                " -- TOO MANY WORDS FOR FKVAL
                TYPE %"RETURN)%" TO GET FIRST 14 WORDS")
              (SETQ FKWORDS 14)))
          (RPTQ FKWORDS
            (SETQ FKRESULT
              (CONS [COND
                (FKREAL (FKFLOAT (IPLUS FKBP RPTN
                  -1)))
                ((OPENR (IPLUS FKBP RPTN -1)
                  FKRESULT)
                  (FKREAL (SETQ FKRESULT (FKFLOAT FKBP)))
                  ((SETQ FKRESULT (OPENR FKBP)
                    (FKACSRETURN FKACS)
                    (RETURN FKRESULT]))
```

{79}

```

(FKSETVAL
  [NLAMBDA (FKADR FKBIAS FKVAL)
    (* NOBIND
      "24-Nov-78 13:00")
    (PROG (FKHNDL FKHT FKACS FKBP FKRESULT)
      (OR FORKDATA (NOFORK))
      (SETQ FKHNDL (FKHNDL FORKDATA))
      (FKWAIT FKHNDL)
      (SETQ FKHT (FKHT FORKDATA))
      (SETQ FKACS (FKACS))
      (SETQ FKBP (LOGOR -29460791296 (IPLUS (LOC FKACS)
        2)))
      (* FULL WORD POINTER TO
        FIRST WORD OF FKACS)
      (* ARGUMENT BITS)
      (FKIDPB 8685804397 FKBP)
      (FKIDPB (LOGOR -19595788288 (FKSYM (QUOTE FKSETV)
        FKHT))
        FKBP)
      (FKIDPB (IPLUS -1 (EVAL FKBIAS)
        (FKSYM FKADR FKHT))
        FKBP)
      (* ADDRESS OF COMMON OR
        VARIABLE)
      (SETQ FKVAL (EVAL FKVAL))
      (OR (LISTP FKVAL)
        (SETQ FKVAL (LIST FKVAL)))
      (COND
        ((IGREATERP (LENGTH FKVAL)
          12)
          (HELP (LENGTH FKVAL)
            " -- TOO MANY WORDS FOR FKSETVAL
            TYPE %"RETURN)%" TO SET FIRST 12 WORDS")
          (SETQ FKVAL (COPY FKVAL))
          (RPLACD (NTH FKVAL 12)
            NIL)))
      (FKIDPB (LENGTH FKVAL)
        FKBP)
      [MAPC FKVAL (FUNCTION (LAMBDA (VAL)
        (FKIDPB (COND
          ((NUMBERP VAL)
            VAL)
          ((NULL VAL)
            0)
          ((EQ VAL T)
            -1)
          (T (ERROR VAL
            " -- NON-NUMERIC ARG IN FKSETVAL"))))
        FKBP]
      (FKSACS FKHNDL FKACS)
      (FKSW FKHNDL 5)
      (FKACSRETURN FKACS)
      (RETURN FKVAL])

```

(FKSYM
[LAMBDA (ID FKHT NOBREAK)

(* LOOKS FOR ID IN FORK HASH TABLE.
IF CANNOT FIND, THEN GOES TO FORK DDT TO LOOK IT
UP.)

```

[OR FKHT (SETQ FKHT (FKHT (OR FORKDATA (NOFORK]
(OR (FIXP ID)
  (GETHASH ID FKHT)
  (PROG (P FKHNDL)

```

```

(* GETS DEFINITION OF ID
FROM DDT FOR THE
FORKNAME)

```

```

  (SETQ P (FKSYMACS FORKDATA))
  (SETQ FKHNDL (FKHNDL FORKDATA))
  (SETA P 1 (GETRADIX50 ID))
  (FKSACS FKHNDL P)
  (FKSW FKHNDL 4)
  (FKRACS FKHNDL P)
  [AND (ZEROP (ELT P 1))
    (COND
      (NOBREAK (RETURN NIL))
      (T (ERROR ID " NOT DEFINED IN FORK"))
    )
  (RETURN (FKSYMPUT FKHT ID (ELT P 2))

```

[81]

```

(FKSYMPUT
  [LAMBDA (FKHT ID V)
    (PROG ((HTL (LIST FKHT)))
      (PUTHASH ID V HTL)

      (FKHT_ FORKDATA (CAR HTL))
      (RETURN V))

```

```

(* EXPANDS HT IF
NECESSARY)

```

[82]

```

(FKSYMP
  [LAMBDA (ID)
    (FKSYM ID NIL T))

```

[83]

```

(GETRADIX50
  [LAMBDA (S)
    (PROG (RADTMP [LEN (COND
      ((ILESSP 6 (NCHARS S))
        6)
      ((NCHARS S]
        (RAD 0)
        (TS (SUBSTRING S 1 -1)))
    [RPTQ LEN (PROGN (SETQ RADTMP (CHCON1 (GNC TS)))
      (COND
        ((AND (IGREATERP RADTMP 57Q)
          (ILESSP RADTMP 72Q))
          (SETQ RADTMP (IDIFFERENCE RADTMP 57Q)))
        ((AND (IGREATERP RADTMP 100Q)
          (ILESSP RADTMP 133Q))
          (SETQ RADTMP (IDIFFERENCE RADTMP 66Q)))
        ((AND (IGREATERP RADTMP 140Q)
          (ILESSP RADTMP 173Q))
          (SETQ RADTMP (IDIFFERENCE RADTMP 126Q)))
        ((EQ RADTMP 56Q)
          (SETQ RADTMP (IDIFFERENCE RADTMP 11Q)))

```

```

((OR (EQ RADTMP 44Q)
      (EQ RADTMP 45Q))
  (SETQ RADTMP (IPLUS RADTMP 2)))
(T (RETURN 0)))
(SETQ RAD (IPLUS (ITIMES RAD 50Q)
                 RADTMP])

```

```

(RETURN RAD])

```

[84]

```

(FKTIME
[LAMBDA (FKEXPR)
  (PROG (FKRESULT FKLISPTIME FKFORKTIME FKHNDL)
    (AND FORKDATA (FKJSYS 15Q (SETQ FKHNDL (FKHNDL FORKDATA)))
      (SETQ FKFORKTIME FKJSYSAC1))
    (FKJSYS 15Q 400000Q)
    (SETQ FKLISPTIME FKJSYSAC1)
    (SETQ FKRESULT (EVAL FKEXPR))
    (FKJSYS 15Q 400000Q)
    (SETQ FKLISPTIME (FQUOTIENT (IDIFFERENCE FKJSYSAC1 FKLISPTIME)
                                FKJSYSAC2))
    (AND FKFORKTIME (FKJSYS 15Q FKHNDL)
      (SETQ FKFORKTIME (FQUOTIENT (IDIFFERENCE FKJSYSAC1
                                                FKFORKTIME)
                                FKJSYSAC2)))
    (RETURN (LIST FKRESULT (FPLUS FKLISPTIME FKFORKTIME)
                  FKLISPTIME FKFORKTIME]))

```

[85]

```

(FKJSYS
[LAMBDA (FKJSYSNO ARG1 ARG2 ARG3 ARG4 ARG5) (* NOBIND
                                              "29-Nov-78 17:38")
  (ASSEMBLE NIL
    (CQ (VAG FKJSYSNO))
    (HRRM 1 , FKJSYS)
    (MOVEI 1 , 4)
    (MOVEM 1 , RETCNT)
    (CQ (FKJSYSARG ARG2))
    (MOVEM 1 , AC2)
    (CQ (FKJSYSARG ARG3))
    (MOVEM 1 , AC3)
    (CQ (FKJSYSARG ARG4))
    (MOVEM 1 , AC4)
    (CQ (FKJSYSARG ARG5))
    (MOVEM 1 , AC5)
    (CQ (FKJSYSARG ARG1))
    (MOVE 2 , AC2)
    (MOVE 3 , AC3)
    (MOVE 4 , AC4)
    (MOVE 5 , AC5)
  FKJSYS
    (JSYS 0)
    (SOS RETCNT)
    (SOS RETCNT)
    (SOS RETCNT)
    (MOVEM 2 , AC2)
    (MOVEM 3 , AC3)

```

```

[ CQ (SETQ FKJSYSAC1 (LOC (AC)
(MOVE 1 , AC2)
[ CQ (SETQ FKJSYSAC2 (LOC (AC)
(MOVE 1 , AC3)
[ CQ (SETQ FKJSYSAC3 (LOC (AC)
(MOVE 1 , RETCNT)
(FASTCALL MKN)
(JRST RETURN)

```

```

RETCNT
(0)
AC2 (0)
AC3 (0)
AC4 (0)
AC5 (0)
RETURN])

```

[86]

```

(FKJSYSARG
[LAMBDA (X)

```

```

(* NOBIND
"29-Nov-78 17:44")

```

```

(PROG (ARG S)
A [SETQ ARG (COND
((NULL X)
0)
((STRINGP X)
[SETQ FKJSYSTR (COND
((ZEROP (CHCON1 (NTHCHAR X -1)))
X)
(T (CONCAT X (CHARACTER 0]
(SETQ S (IPLUS (LOC (CAR FKJSYSTR))
(LSH (LOGAND (LOC (CDR FKJSYSTR))
7)
18)))
(LOGOR (IQUOTIENT S 5)
117440512
(LLSH (IDIFFERENCE 36 (ITIMES (IREMAINDER S 5)
7))
30)))
((ARRAYP X)
(IPLUS 2 (LOC X)))
((NUMBERP X)
X)
(T (SETQ X (ERROR X "FKJSYS ARGUMENT ERROR"))
(GO A]
(RETURN (VAG ARG])

```

[87]

```

(FKWAIT
[LAMBDA (FKHNDL)

```

```

(* NOBIND
"24-Nov-78 16:42")

```

```

(PROG NIL
WAIT(FKJSYS 156Q FKHNDL)
(SETQ FKSTATUS (LRSH FKJSYSAC1 22Q))
(COND
((EQ FKSTATUS 2))

```

```

(* RFSTS)

```

```

      ((EQ FKSTATUS 1)
      (FKJSYS 206Q FKHNDL) (* GPJFN)
      (FKJSYS 104Q (LOGAND FKJSYSAC2 777777Q))
      (* DOBE)
      (FKJSYS 162Q FKHNDL) (* HFORK)
      (DISMISS 12Q))
      ((MEMB FKSTATUS (QUOTE (0 4 5)))
      (DISMISS 764Q)
      (GO WAIT))
      (T (HELP "UNUSUAL FORK STATUS:" FKSTATUS1)
    )

```

```

(ADDTTOVAR GLOBALVARS FORKDATA DSPNOWAITFLG)

```

```

(ADDTTOVAR AFTERSYSOUTFORMS (PROGN (RPLACA FORKDATA NIL)
                                     (FKKILL)))

```

```

(AND (EQ (EVALV (QUOTE FORKDATA))
        (QUOTE NOBIND))
      (SETQ FORKDATA NIL))

```

```

(PUTPROPS FKIDPB MACRO [(E P)
                        (ASSEMBLE NIL
                          (CQ (VAG E))
                          (PUSHN 1)
                          (CQ P)
                          (POPN 2)
                          (IDPB 2 , 0 (1))

```

```

(PUTPROPS FKRACS MACRO ((FKHNDL A)
                        (ASSEMBLE NIL
                          (CQ (VAG (IPLUS (LOC A)
                                           2)))
                          (PUSHN 1)
                          (CQ (VAG FKHNDL))
                          (POPN 2)
                          (JSYS 161Q))
                        (* RFACS)
    ))

```

```

(PUTPROPS FKSACS MACRO ((FKHNDL A)
                        (ASSEMBLE NIL
                          (CQ (VAG (IPLUS (LOC A)
                                           2)))
                          (PUSHN 1)
                          (CQ (VAG FKHNDL))
                          (POPN 2)
                          (JSYS 160Q))
                        (* SFACS)
    ))

```

```

(PUTPROPS PUTTYP MACRO [(N)
                        (ASSEMBLE NIL
                          (CQ FKCBP)
                          (HRRZI 2 , N)
                          (IDPB 2 , 0 (1))

```

```

(PUTPROPS FKHNDL MACRO ((X)

```

```

(CAR X)))

(PUTPROPS FKHT MACRO ((X)
  (CAADR X)))

(PUTPROPS FKSHR MACRO ((X)
  (CADDR X)))

(PUTPROPS FKSYMACS MACRO ((X)
  (CADR (CDDDR X))))

(PUTPROPS FKDDT MACRO ((X)
  (CADR (CDDDDR X))))

(PUTPROPS FKJFN MACRO ((X)
  (CADDR (CDDDDR X))))

(PUTPROPS FKHT_ MACRO ((X Y)
  (RPLACA (CADR X)
    Y)))

(PUTPROPS FKDDT_ MACRO ((X Y)
  (RPLACA (CDDR (CDDDR X))
    Y)))

(PUTPROPS FKPROG MACRO ((X)
  (CADDR (CDDDDR X))))

(PUTPROPS FKHALT MACRO ((X)
  (CADADR X)))

(RPAQQ FORKBLOCKS ((FKCALLBLOCK FKCALL FKCATYPE FKSR SAILCALL SAILARG
  SAILSTRING FKACS FKACSRETURN FKRTN
  (NOLINKFNS . T)
  (ENTRIES FKCALL SAILCALL FKCATYPE FKACS
    FKACSRETURN))
  (FKARRAYBLOCK FKARRAY FKCORGET FKELT FKELTI FKELTR FKSETA
    FKARRAYP FKBCHECK FKARRADR FKFLOAT (NOLINKFNS . T)
    (ENTRIES FKARRAY FKELT FKELTI FKELTR FKSETA
      FKARRAYP FKFLOAT))
  (FKSYMBLOCK FKSYP FKSYP FKSYP GETRADIX50 (NOLINKFNS . T)
    (ENTRIES FKSYP FKSYP FKSYP))
  (FKJSYSBLOCK FKJSYS FKJSYSARG (NOLINKFNS . T)
    (ENTRIES FKJSYS)))
[DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK: FKCALLBLOCK FKCALL FKCATYPE FKSR SAILCALL SAILARG SAILSTRING
  FKACS FKACSRETURN FKRTN (NOLINKFNS . T)
  (ENTRIES FKCALL SAILCALL FKCATYPE FKACS FKACSRETURN))
(BLOCK: FKARRAYBLOCK FKARRAY FKCORGET FKELT FKELTI FKELTR FKSETA
  FKARRAYP FKBCHECK FKARRADR FKFLOAT (NOLINKFNS . T)
  (ENTRIES FKARRAY FKELT FKELTI FKELTR FKSETA FKARRAYP FKFLOAT))
(BLOCK: FKSYMBLOCK FKSYP FKSYP FKSYP GETRADIX50 (NOLINKFNS . T)
  (ENTRIES FKSYP FKSYP FKSYP))
(BLOCK: FKJSYSBLOCK FKJSYS FKJSYSARG (NOLINKFNS . T)
  (ENTRIES FKJSYS))
]
(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS

```

(ADDTOVAR NLANA SAILCALL FKCALL)

(ADDTOVAR NLAML FKSETVAL FKVALI FKVALR FKVAL FKSETA FKELTR FKELTI FKELT
FKARRAY FKX FKINIT)

)

(DECLARE: DONTCOPY

(FILEMAP (NIL (1295 37331 (FKINIT 1307 . 5060) (FKKILL 5064 . 5982) (FKSAVE 5986 . 6375) (FKDDT 6379 . 7661) (FKCALL 7665 . 11545) (FKCATYPE 11549 . 11870) (FKSR 11874 . 12612) (SAILCALL 12616 . 15106) (SAILARG 15110 . 18208) (SAILSTRING 18212 . 18853) (FKACS 18857 . 19041) (FKACSRETURN 19045 . 19159) (FKRTN 19163 . 19479) (NOFORK 19483 . 19606) (FKCALLERR 19610 . 19701) (FKSW 19705 . 20590) (FKX 20594 . 20686) (FKTTYSET 20690 . 21667) (FKARRAY 21671 . 24128) (FKCORGET 24132 . 24466) (FKELT 24470 . 24708) (FKELTI 24712 . 25073) (FKELTR 25077 . 25443) (FKSETA 25447 . 25851) (FKARRAYP 25855 . 26061) (FKARRAYSIZE 26065 . 26135) (SAILARRAYSIZE 26139 . 26444) (FKARRAYTYPE 26448 . 26722) (FKBCHECK 26726 . 26845) (FKARRADR 26849 . 28129) (FKFLOAT 28133 . 28250) (ARRLOC 28254 . 28406) (FKVAL 28410 . 28571) (FKVALR 28575 . 28689) (FKVALI 28693 . 30236) (FKSETVAL 30240 . 31842) (FKSYM 31846 . 32550) (FKSYMPUT 32554 . 32763) (FKSYMP 32767 . 32813) (GETRADIX50 32817 . 33686) (FKTIME 33690 . 34359) (FKJSYS 34363 . 35888) (FKJSYSARG 35892 . 36645) (FKWAIT 36649 . 37328))))))

STOP

(FILECREATED " 7-Aug-79 19:03:11" <RBECHTAL>HASHER..38 8393

changes to: GETSH

previous date: " 6-Aug-79 14:37:00" <RBECHTAL>HASHER..37)

(PRETTYCOMPRINT HASHERCOMS)

(RPAQQ HASHERCOMS ((VARS * HASHERVARS)
(FNS * HASHERFNS)))

(RPAQQ HASHERVARS (MEMORY MEMSIZE MEMLIMIT MEMFACTOR))

(RPAQQ MEMORY NIL)

(RPAQQ MEMSIZE 256)

(RPAQQ MEMLIMIT 0)

(RPAQQ MEMFACTOR 0)

(RPAQQ HASHERFNS (ADDH CREATH FASTHAK GETH GETSH GETSTRIP LOCH MAPH
MEMDENSITY MEMTEST NEWHASH NEXTH PREHASH PUTH
PUTSH))

(DEFINEQ

[88]

(ADDH
[LAMBDA (ARGS NEWVAL)

(* edited:
" 6-Aug-79 13:51")

(* ADDH is really no longer necessary -
PUTSH does the same job now.)

(PUTSH ARGS NEWVAL))

[89]

(CREATH
[LAMBDA (SIZE)

(* edited:
" 6-Aug-79 13:58")

(* CREATH creates an array, called MEMORY, of the
size specified by the argument given to CREATH.
This array will be treated as a hash array, and is
used to store the assertion retrieval information.)

(SETQ MEMORY (ARRAY SIZE))

(SETQ MEMSIZE SIZE)

(SETQ MEMFACTOR (ADD1 (IQUOTIENT (LOG SIZE)
5)))

(SETQ MEMFILLED 0)

```
(SETQ MEMFULLSIZE (IQUOTIENT (ITIMES SIZE 4)
                               51))
```

[90]

```
(FASTHAK
  [LAMBDA NIL
```

```
(* edited:
" 6-Aug-79 13:59" |
(* FASTHAK provides a |
way to look at the |
contents of MEMORY.) |
```

```
(MAPH MEMORY MEMSIZE (FUNCTION MEMTEST))
```

[91]

```
(GETH
  [LAMBDA (ARGS)
```

```
(* edited:
" 6-Aug-79 14:01" |
|
|
|
|
```

```
(* GETH retrieves the CDR of the element of the
array whose CAR contains ARGS.)
```

```
(ELTD MEMORY (LOCH ARGS])
```

[92]

```
(GETSH
  [LAMBDA (ARGS)
```

```
(* edited:
" 7-Aug-79 19:03")
```

```
(* GETSH is insured to return a stream.
It's not unlike GETH, but will create
(and store) a stream if necessary.)
```

```
(* The APPEND is |
necessary because |
RETSTREAM reuses a |
scratchlist) |
```

```
(OR (GETH ARGS)
    (PUTH (APPEND ARGS)
          (NEWSTREAM]))
```

[93]

```
(GETSTRIP
  [LAMBDA (ARGS)
```

```
(* edited:
" 6-Aug-79 14:04" |
(* GETSTRIP returns a |
list of assertions which |
match ARGS.) |
```

```
(STRIPSTREAM (GETH ARGS])
```

[94]

```
(LOCH
  [LAMBDA (ARGS PUTFLG)
```

```
(* edited:
  " 6-Aug-79 14:08")
```

```
(* LOCH generates a location in MEMORY whose CAR is
  ARGS. This may involve moving down in the case of
  collisions. PUTFLG signals that the indexing is for
  insertion (so that erased locations can be reused).)
```

```
(PROG (LOC CONT)
  (SETQ LOC (PREHASH ARGS))
  (SETQ MEMTESTCNT 1)
  GLOOP
    (SETQ CONT (ELT MEMORY LOC))
    (COND
      ((OR (EQUAL CONT ARGS)
            (NULL CONT)
            (AND PUTFLG (EQ CONT (QUOTE *erased*))
                  (RETURN LOC)))
       (T (SETQ LOC (NEXTH LOC ARGS))
           (SETQ MEMTESTCNT (ADD1 MEMTESTCNT))
           (GO GLOOP))
```

[95]

```
(MAPH
  [LAMBDA (ARY ARYSZ ARYFN)
```

```
(* edited:
  " 6-Aug-79 14:09")
```

```
(* MAPH maps ARYFN, a function of two arguments,
  over the array ARY, which has size ARYSZ.
  Used by MEMTEST and NEWHASH.)
```

```
(PROG ((COUNT 1)
  CONTENT)
  MPLOOP
    (COND
      ((GREATERP COUNT ARYSZ)
       (RETURN)))
    [COND
      ((ELT ARY COUNT)
       (APPLY* ARYFN (ELT ARY COUNT)
                    (ELTD ARY COUNT)
                    (SETQ COUNT (ADD1 COUNT))
                    (GO MPLOOP))
```

[96]

```
(MEMDENSITY
  [LAMBDA NIL
```

```
(* edited:
  " 6-Aug-79 14:22")
(* MEMDENSITY calculates
  how full MEMORY is.)
```

```

(PRIN1 "memory is ")
(PRIN1 (FTIMES 100.0 (FQUOTIENT MEMFILLED MEMSIZE)))
(PRIN1 " percent full.")
(TERPRI)
(PRIN1 MEMFILLED)
(PRIN1 " out of ")
(PRIN1 MEMSIZE)
(PRIN1 " spaces are in use.")
(TERPRI)

```

[97]

```

(MEMTEST
 [LAMBDA (X Y)

```

```

(* edited:
 " 6-Aug-79 14:29")

```

```

(* MEMTEST prints useful information about the
 contents of MEMORY. However, if there's a lot in
 memory, it gets very dull.)

```

```

(COND
 ((NULL X))
 (T (PRIN1 COUNT)
      (PRIN1 " ")
      (PRIN1 X)
      (PRIN1 " ")
      (PRIN1 (CAAR Y))
      (PRIN1 " ")
      (PRIN1 (LENGTH (CADR Y)))
      (TERPRI))

```

[98]

```

(NEWHASH
 [LAMBDA NIL

```

```

(* edited:
 " 6-Aug-79 14:30")
(* NEWHASH creates a new
 hash array. Effectively
 a dynamic expansion of
 MEMORY.)

```

```

(PROG ((A MEMORY)
      (OLDSIZE MEMSIZE))
 (CREATH (PLUS MEMSIZE (IQUOTIENT MEMSIZE 2)))
 (MAPH A OLDSIZE (FUNCTION (LAMBDA (LEFT RIGHT)
      (COND
        [(OR (NULL LEFT)
              (EQ LEFT (QUOTE *erased*)]
          (T (PUTH LEFT RIGHT))

```

[99]

```

(NEXTH
 [LAMBDA (LOC ARG)

```

```

(* edited:
 " 6-Aug-79 14:32")

```

```

(* NEXTH generates a new address in the case of

```

collisions. It's a simple "move down", with the increment selected to be relatively prime to an reasonable array size (preventing wraparound), and widely spaced.)

```
(PROG (NEWLOC)
  (SETQ NEWLOC (IDIFFERENCE LOC 659))
  NXTLP
  (COND
    ((GREATERP 1 NEWLOC)
      (SETQ NEWLOC (IPLUS MEMSIZE NEWLOC))
      (GO NXTLP))
    (T (RETURN NEWLOC)))
```

[100]

```
(PREHASH
  [LAMBDA (L)
```

```
(* edited:
  " 6-Aug-79 14:33")
```

(* PREHASH generates an address given a retrieval spec. It is the primary hashing function.)

```
(PROG (C N)
  (SETN PREHASHSUM 0)
  OUTER
  (SETN PREHASHSUM1 0)
  (SETQ N 3)
  INNER
  [COND
    ((NULL L)
      (RETURN (ADD1 (IREMAINDER (IPLUS PREHASHSUM PREHASHSUM1)
        MEMSIZE)))
      (SETQ C (CAR L))
      (SETN PREHASHSUM1 (IPLUS (LSH PREHASHSUM1 8)
        (LOGAND (COND
          ((LITATOM C)
            (LOC C))
          ((NUMBERP C)
            (LOC (VAG C)))
          ((STRINGP C)
            (LOC (MKATOM C)))
          ((LISTP C)
            (PREHASH C))
          (T (HELP
              "BAD ARG - PREHASH"))))
          255)))
      (SETQ L (CDR L))
      (SETQ N (SUB1 N))
      (COND
        ((ZEROP N)
          (SETN PREHASHSUM (IPLUS PREHASHSUM PREHASHSUM1))
          (GO OUTER)))
        (T (GO INNER]))
```

[101]

```
(PUTH
  [LAMBDA (ARGS AVAL)
```

```
(* edited:
" 6-Aug-79 14:34" |
(* PUTH sticks things in
MEMORY, expanding if |
necessary.) |
```

```
(COND
  ((IGREATERP MEMFILLED MEMFULLSIZE)
   (NEWHASH)))
(SETQ MEMFILLED (ADD1 MEMFILLED))
(PROG ((LOC (LOCH ARGS T)))
  (SETA MEMORY LOC ARGS)
  (SETD MEMORY LOC AVAL)
  (RETURN AVAL))
```

[102]

```
(PUTSH
  [LAMBDA (ARGS AVAL)
```

```
(* edited:
" 6-Aug-79 14:36" |
(* PUTSH places AVAL in
the stream associated |
with ARGS) |
```

```
(PUTSTREAM (GETSH ARGS)
  AVAL])
```

)

```
(DECLARE: DONTCOPY
```

```
(FILEMAP (NIL (555 8369 (ADDH 567 . 852) (CREATH 856 . 1477) (FASTHAK
1481 . 1766) (GETH 1770 . 2069) (GETSH 2073 . 2548) (GETSTRIP 2552 .
2820) (LOCH 2824 . 3617) (MAPH 3621 . 4226) (MEMDENSITY 4230 . 4790) (
MEMTEST 4794 . 5326) (NEWHASH 5330 . 5859) (NEXTH 5863 . 6522) (PREHASH
6526 . 7616) (PUTH 7620 . 8085) (PUTSH 8089 . 8366))))
STOP
```

(FILECREATED "28-Aug-79 21:06:16" <RBECHTAL>INTERP..37 12152

changes to: JUSTBUILD

previous date: "27-Aug-79 21:31:58" <RBECHTAL>INTERP..36)

(PRETTYCOMPRINT INTERPCOMS)

(RPAQQ INTERPCOMS [(FNS * INTERPFNS)
 (VARS (VDRELS (QUOTE (LESS-THAN SAME-AS GREATER-THAN)
)

(RPAQQ INTERPFNS (ANDHACK APPLYRULE CONSTRUCT GETPULSAR JUSTBUILD
 MESSAGE1 NOTHACK ORACLEHACK ORBUILD ORHACK
 SAVEPULSAR SWEEPER UNLESSHACK VAR?))

(DEFINEQ

[103]

(ANDHACK
 [LAMBDA (CONDITIONS ACTIONS EV)

(* edited:
 " 7-Aug-79 10:12")

(* ANDHACK handles anded conditions
(those without special modifiers). An AND is true
(succeeds) if the confidence in the assertion it
finds is greater than .1 (an arbitrary threshold).
Like all of the other hacks, ANDHACK relies on
MAPRETRIEVE to do the real work.
Oracles are evaluated first, so that their results
will exist in the network for the MAPRETRIEVE to
find.)

(ORACLEHACK (CAR CONDITIONS))
(MAPRETRIEVE (CAR CONDITIONS)
 (LIST (CDR CONDITIONS)
 ACTIONS EV)
 (FUNCTION (LAMBDA (X P)
 (PROG ((CLIST (CAR P))
 (ACTIONS (CADR P))
 (EV (CADDR P)))
 (COND
 ((GREATERP (GETCON X)
 .1)
 (SWEEPER CLIST ACTIONS (CONS X EV))
 (RETURN T))

[104]

(APPLYRULE
 [LAMBDA (RULENAME PREBIND)

(* edited:
 "19-Jul-79 14:43")

(* APPLYRULE is the function that starts all of the

work of the rule interpreter.
 When APPLYRULE is called on a rule, it starts the process of mapping retrieval functions over the data base based on the conditions of the rule. Binding of variables is accomplished in this version by rewriting the remaining conditions with the bindings substituted for the variables.)

```
(SWEEPER (SUBLIS PREBIND (GETPROP RULENAME (QUOTE CONDITIONS))) |
  (SUBLIS PREBIND (GETPROP RULENAME (QUOTE ACTIONS))) |
  (CONS RULENAME))
```

[105]

```
(CONSTRUCT
  [LAMBDA (ACTIONS EV COUNT)
```

```
(* edited:
  "27-Aug-79 11:47")
```

```
(PROG (FIRST)
```

(* CONSTRUCT is the function that steps through the actions of a rule and passes them to the appropriate conclusion building functions.)

```
(OR COUNT (SETQ COUNT 1)) |
CLOOP
(COND
  ((NULL ACTIONS)
   (RETURN T))
  (T (SETQ FIRST (CAR ACTIONS))
    (COND
      ((EQ (CAR FIRST)
            (QUOTE *OR*))
       (ORBUILD (CDR FIRST)
                 EV))
      ((EQ (CAR FIRST)
            (QUOTE *REPORT*))
       (SETQ RESULTLIST (CONS (MESSAGE1 FIRST)
                              RESULTLIST)))
      (T (JUSTBUILD FIRST EV COUNT))) |
  (SETQ ACTIONS (CDR ACTIONS))
  (GO CLOOP])
```

[106]

```
(GETPULSAR
  [LAMBDA (NODE)
```

```
(* edited:
  " 7-Aug-79 10:14")
```

(* GETPULSAR, as its name suggests, gets the pulsar (if any) associated with a node. Isolating this as a function allows redesign of pulsar storage with minimal rewriting.)

```
(GETPROP NODE (QUOTE PULSAR])
```

[107]

(JUSTBUILD

[LAMBDA (SPEC EV NUMBER)

(* edited:

"28-Aug-79 21:06")

(PROG (NEWNODE NEWFLG MASSAGESPEC)

(* This is the function that actually builds conclusions in the assertion memory, or data base. Because of the immediacy of the stream coroutines, it is necessary to build the derivation tree before actually adding the new assertion to the memory, lest the new assertion be used for some rule without having its confidence calculable.)

(SETQ MASSAGESPEC (MESSAGE1 SPEC))

[SETQ NEWNODE (COND

((CAR (GETSTRIP MASSAGESPEC)))

(T (SETQ NEWFLG (GENSYM)

(SETQ RESULTLIST (CONS NEWNODE RESULTLIST))

[PUTPROP NEWNODE (QUOTE DERIVE*)

(CONS (REVERSE EV)

(GETPROP NEWNODE (QUOTE DERIVE*)

[COND

(NEWFLG (SET NEWNODE MASSAGESPEC))

(SETQ ASSERTIONS (CONS NEWNODE ASSERTIONS))

(SAVEPULSAR NEWNODE)

(SERT (MESSAGE1 SPEC)

NEWNODE))

(T (PULSE (GETPULSAR NEWNODE)

(RETURN NEWFLG])

[108]

(MESSAGE1

[LAMBDA (SPEC LIST)

(* edited:

" 7-Aug-79 10:17")

(* MESSAGE1 takes a condition (or action) and binds its variables to their interpreter values. GETMRVAL (called by MESSAGE1) has since been extended to deal with lists as well as atoms, so that calls to MESSAGE1 could be directly replaced with calls to GETMRVAL.)

(MAPCAR SPEC LIST (FUNCTION (LAMBDA (X)

(COND

((VAR? X)

(GETMRVAL X))

(T X])

[109]

```
(NOTHACK
  [LAMBDA (CONDITIONS ACTIONS EV)
```

```
(* edited:
  " 7-Aug-79 10:18")
```

```
(* NOTHACK requires a confidence less than -.1 to
  continue the rule evaluation.
  General comments about the connective hacks apply.)
```

```
(ORACLEHACK (CADAR CONDITIONS))
(MAPRETRIEVE (CADAR CONDITIONS)
  (LIST (CDR CONDITIONS)
    ACTIONS EV)
  (FUNCTION (LAMBDA (X P)
    (PROG ((CLIST (CAR P))
      (ACTIONS (CADR P))
      (EV (CADDR P)))
    (COND
      ((LESSP (GETCON X)
        -.1)
        (SWEEPER CLIST ACTIONS
          (CONS (LIST (QUOTE NOT)
            X)
            EV))
        (RETURN T))
```

[110]

```
(ORACLEHACK
  [LAMBDA (SPEC)
```

```
(* edited:
  " 7-Aug-79 10:21")
```

```
(* ORACLEHACK deals with oracular conditions.
  Firsts, it tests for the presence of an oracle.
  If one is found, it then checks the data base to see
  if it has already been computed.
  If not, it computes the oracle, and places the
  result in the memory, where it can be used by the
  normal condition evaluation procedure.
  Computation is restricted to oracles with LASTARG on
  their property list under the property name ORTYPE.
  Such oracles bind their last argument.)
```

```
(PROG (PTR LAST-ARG LASTCONS ANS ORTYPE)
  (COND
    ((GETPROP (CAR SPEC)
      (QUOTE ORACLE))
      (SETQ SPEC (GETMRVAL SPEC T))
      (SETQ ORTYPE (GETPROP (CAR SPEC)
        (QUOTE ORTYPE)))
      (SELECTQ ORTYPE
        [LASTARG (SETQ LASTCONS (LAST SPEC))
          (SETQ PTR (NLEFT SPEC 2))
          (SETQ LAST-ARG (CADR PTR))
```


[113]

```
(SAVEPULSAR
  [LAMBDA (NODE)
```

```
(* edited:
  " 7-Aug-79 10:25")
```

```
(* SAVEPULSAR saves a pulsar on a node.
  If the method of storing pulsars should change, the
  modularity of SAVEPULSAR and GETPULSAR insure that
  they are the only functions that need to be changed,
  since all pulsar access is done through them.)
```

```
(PUTPROP NODE (QUOTE PULSAR)
  (PULSAR])
```

[114]

```
(SWEEPER
  [LAMBDA (CONDITIONS ACTIONS EV)
```

```
(* edited:
  "27-Aug-79 21:31")
```

```
(PROG (THISCOND C)
```

```
(* SWEEPER, which is where the work used to get
  done, now is merely a big switch which determines
  the appropriate condition handling function to call.
  The functions generated by these condition handlers
  and FUNCTIONWRITER will then call SWEEPER
  recursively.)
```

```
[COND
  ((NULL CONDITIONS)
    (RETURN (CONSTRUCT ACTIONS EV)))
  (T (SETQ THISCOND (CAAR CONDITIONS)
    [COND
      ((MEMB THISCOND VDRELS)
        (AND (APPLY THISCOND (GETMRVAL (CDAR CONDITIONS)))
          (SWEEPER (CDR CONDITIONS)
            ACTIONS EV)))
      ((EQ THISCOND (QUOTE *OR*))
        (ORHACK CONDITIONS ACTIONS EV))
      [(EQ THISCOND (QUOTE *NOT*))
        (SETQ C (CADAR CONDITIONS))
        (COND
          ((MEMB (CAR C)
            VDRELS)
            (OR (APPLY (CAR C)
              (GETMRVAL (CDR C)))
              (SWEEPER (CDR CONDITIONS)
                ACTIONS EV)))
          (T (NOTHACK CONDITIONS ACTIONS EV))
        (EQ THISCOND (QUOTE *UNLESS*))
        (UNLESSHACK CONDITIONS ACTIONS EV))
        (T (ANDHACK CONDITIONS ACTIONS EV))
```

[115]

(UNLESSHACK

[LAMBDA (CONDITIONS ACTIONS EV)

(* edited:
" 7-Aug-79 10:30")

(* UNLESSHACK is the connective hack that deals with UNLESS conditions. While the general comments about hacks apply to UNLESSHACK, it is quite different in that it expects to find its assertion in the memory, adding it (with confidence 0.0) if necessary. Since UNLESSes succeed if the confidence in their assertion is 0.0 or less, adding an assertion to memory in UNLESS forces the condition to succeed. UNLESSes are only blocked if information (with positive confidence) already exists in the network. Given the parallel rule application, this feature can create problems unless care is taken in rule construction. This problem will be discussed further in later working papers.)

(ORACLEHACK (CADAR CONDITIONS))

(COND

[(STRIPSTREAM (RETSTREAM (CADAR CONDITIONS)

(T (CASSERT (MESSAGE1 (CADAR CONDITIONS))
0.0)))

(MAPRETRIEVE (CADAR CONDITIONS)

(LIST (CDR CONDITIONS)
ACTIONS EV)

(FUNCTION (LAMBDA (X P)

(PROG ((CLIST (CAR P))
(ACTIONS (CADR P))
(EV (CADDR P)))

(COND

((LEQ (GETCON X)
0.0)

(SWEEPER CLIST ACTIONS

(CONS (LIST (QUOTE UNLESS)
X)

EV))

(RETURN T))

[116]

(VAR?

[LAMBDA (Q)

(* edited:
"11-May-79 01.46")

(* VAR? tests to see if an atom is in variable format. Variables start with an asterisk.)

(EQ (CHCON1 Q)
42))

)

(RPAQQ VDRELS (LESS-THAN SAME-AS GREATER-THAN))

(DECLARE: DONTCOPY

(FILEMAP (NIL (462 12077 (ANDHACK 474 . 1502) (APPLYRULE 1506 . 2184) (CONSTRUCT 2188 . 2953) (GETPULSAR 2957 . 3386) (JUSTBUILD 3390 . 4509) (MESSAGE1 4513 . 5148) (NOTHACK 5152 . 5903) (ORACLEHACK 5907 . 7358) (ORBUILD 7362 . 7898) (ORHACK 7902 . 8464) (SAVEPULSAR 8468 . 8995) (SWEEPER 8999 . 10184) (UNLESSHACK 10188 . 11766) (VAR? 11770 . 12074))))

)
STOP

(FILECREATED " 6-Aug-79 17:32:02" <RBECHTAL>MANIPULATE..20 8876

changes to: ASSERT BUMP CASSERT DENY GETUPLE MATCHER MAYBE
RETRIEVER RETVARS SERT STATE

previous date: "19-Jul-79 15:45:55" <RBECHTAL>MANIPULATE..19)

(PRETTYCOMPRINT MANIPULATECOMS)

```
(RPAQQ MANIPULATECOMS [(FNS * MANIPULATEFNS)
                        (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
                          COMPILERVARS
                          (ADDVARS (NLAMA STATE MAYBE DENY)
                                    (NLAML)
                                    (LAMA)])
```

```
(RPAQQ MANIPULATEFNS (ASSERT BUMP CASSERT DENY GETUPLE MATCHER MAYBE
                           RETRIEVER RETVARS SERT STATE))
(DEFINEQ
```

[117]

```
(ASSERT
  [LAMBDA (ARGLIST NODENAME)
```

```
(* edited:
  " 6-Aug-79 17:03")
```

```
(* ASSERT creates an assertion and places it in
memory, keyed by the appropriate retrieval
specifications. ASSERT makes no statement regarding
the confidence in the assertion, and thus should be
used with the greatest caution, to avoid fouling
rules. ASSERT takes one argument, a list, which it
evaluates.)
```

```
(PROG (REPLY LEN A)
  [COND
    ((GETSTRIP ARGLIST)
     (RETURN (CAR (GETSTRIP ARGLIST))
    [SETQ REPLY (COND
      (NODENAME)
      (T (GENSYM)
    (SETQ ASSERTIONS (CONS REPLY ASSERTIONS))
    (SET REPLY ARGLIST)
    (SAVEPULSAR REPLY)
    (SERT ARGLIST REPLY)
    (RETURN REPLY)])
```

[118]

```
(BUMP
  [LAMBDA (L)
```

```
(* edited:
  " 6-Aug-79 17:04")
```

(* BUMP counts in binary, using a list of Ts and

NILs in place of 1s and 0s. Given a list of Ts and NILs, returns a list of Ts and NILs that is "plus one" of its argument.)

```
(PROG (ANS)
  BLOOP1
    [COND
      ((NULL L)
        (RETURN (DREVERSE ANS)))
      ((CAR L)
        (SETQ ANS (CONS NIL ANS))
        (SETQ L (CDR L))
        (GO BLOOP1))
      (T (SETQ ANS (CONS T ANS))
        (SETQ L (CDR L))
```

```
  BLOOP2
    (COND
      ((NULL L)
        (RETURN (DREVERSE ANS)))
      (T (SETQ ANS (CONS (CAR L)
        ANS))
        (SETQ L (CDR L))
        (GO BLOOP2)))
```

[119]

```
(CASSERT
  [LAMBDA (SPEC VAL)
```

```
(* edited:
  " 6-Aug-79 17:08")
```

(* CASSERT works like ASSERT, only it establishes a confidence in the assertion it creates. CASSERT takes two arguments. The first is taken to be the assertion spec, and the second the confidence in the assertion. If the confidence argument is positive, it is used as the measure of belief in the assertion, and if negative, it is used as the measure of disbelief. Whichever measure is not specified is set to zero.)

```
(PROG (NEWNODE)
  [COND
    ((GETSTRIP SPEC)
      (RETURN (CAR (GETSTRIP SPEC)
        (SETQ NEWNODE (GENSYM))
        (SETQ ASSERTIONS (CONS NEWNODE ASSERTIONS))
        [COND
          ((GREATERP VAL 0.0)
            (PUTPROP NEWNODE (QUOTE MB)
              VAL)
            (PUTPROP NEWNODE (QUOTE MD)
              0.0))
          (T (PUTPROP NEWNODE (QUOTE MB)
            0.0)
            (PUTPROP NEWNODE (QUOTE MD)
```

```
(ABS VAL)
(SET NEWNODE SPEC)
(SAVEPULSAR NEWNODE)
(SERT SPEC NEWNODE)
(RETURN NEWNODE))
```

[120]

```
(DENY
 [NLAMBDA L
```

```
(* edited:
 " 6-Aug-79 17:10")
```

```
(* DENY asserts its argument
(s) (unevaluated) with confidence -1.0.
The most common anticipated use of this function is
at top level (LISP) in APPLY format, e.g. STATE
(PLATFORM CONTACT34).)
```

```
(CASSERT L -1.0))
```

[121]

```
(GETUPLE
 [LAMBDA (ASSER)
```

```
(* edited:
 " 6-Aug-79 17:11")
```

```
(* Given an assertion node, GETUPLE returns the
tuple (content) of that node.)
```

```
(EVAL ASSER))
```

[122]

```
(MATCHER
 [LAMBDA (L1 L2)
```

```
(* edited:
 " 6-Aug-79 17:17")
```

```
(* MATCHER is used by SERT to construct retrieval
specifications from assertion tuples.
MATCHER takes two arguments, an assertion tuple and
a "binary number" list, such as that returned from
BUMP. Wherever the "binary
number" list contains T, the corresponding element of the
assertion tuple is used in the retrieval
specification. Where the BNlist contains NIL, a * is
inserted in the retrieval specification.
The retrieval specification is returned.)
```

```
(PROG (ANS)
 MLOOP
 [COND
 ((NULL L1)
 (RETURN (DREVERSE ANS)))
 ((CAR L1)
```

```

      (SETQ ANS (CONS (CAR L2)
                      ANS)))
    (T (SETQ ANS (CONS (QUOTE *)
                      ANS])
      (SETQ L1 (CDR L1))
      (SETQ L2 (CDR L2))
      (GO MLOOP]))

```

[123]

(MAYBE
[NLAMBDA L

```
(* edited:
" 6-Aug-79 17:17")
(* MAYBE functions like
DENY, only gives a
confidence of 0.0.)
```

(CASSERT L 0.01)

[124]

(RETRIEVER
[LAMBDA (SPEC)

(* edited:
" 6-Aug-79 17:29")

```
(* RETRIEVER is the workhorse function that gets
stuff out of the memory. RETRIEVER takes a single
argument list (evaluated), which should be either an
assertion tuple, or an assertion tuple with
variables in some places. (A variable is an atom
that starts with a star, such as *PLAT.) RETRIEVER
returns a list of answers, each of the form
(assertionnodematched alist), where the alist is a
set of CONSES of the variables together with the
ground instances which they matched.
For example, (RETRIEVER (QUOTE
(SIGHTING *PLAT *SNODE))) might return
((A0034 (*PLAT . CONSOLE) (*SNODE . SIGHTING1))
(A0765 (*PLAT . MINSK) (*SNODE . SIGHTING55)))
```

```

(PROG (RES)
  [MAPC (GETSTRIP (RETVAR$ SPEC))
    (FUNCTION (LAMBDA (W)
      (PROG (RES1)
        [MAP2C SPEC (GETUPLE W)
          (FUNCTION (LAMBDA (A B)
            (COND
              ((VAR? A)
                (SETQ RES1
                  (CONS (CONS A B)
                    RES1)
                (SETQ RES (CONS (CONS W RES1)
                  RES)
                (RETURN RES1))

```

[125]

```
(RETVARS
  [LAMBDA (SPEC)
```

```
(* edited:
" 6-Aug-79 17:30"
(* RETVARS messages
RETRIEVER specs to turn
them into retrieval
specifications.)
```

```
(MAPCAR SPEC (FUNCTION (LAMBDA (ITEM)
  (COND
    ((VAR? ITEM)
     (QUOTE *))
    (T ITEM))
```

[126]

```
(SERT
  [LAMBDA (SPEC NODENAME)
```

```
(* edited:
" 6-Aug-79 16:28")
```

```
(PROG (LEN A)
```

```
(* SERT is the function that actually stores the
assertion nodes in the memory under specifications
generated by MATCHER with the help of BUMP.
In this version, we are not storing under retrieval
specifications where the relation field is
wild-carded. To permit retrieval of assertions with
the relation field variable
(binding the relation field), remove the SUB1 and
the CONS of T at SLOOPJB. Also, remove the ADDH just
before SLOOPJB.)
```

```
(SETQ LEN (SUB1 (LENGTH SPEC)))
(SETQ A NIL)
(RPTQ LEN (SETQ A (CONS NIL A)))
(ADDH (MATCHER (CONS T A)
  SPEC)
  NODENAME)
SLOOPJB
(ADDH (MATCHER (CONS T (SETQ A (BUMP A)))
  SPEC)
  NODENAME)
(COND
  ((MEMB NIL A)
   (GO SLOOPJB)))
(ENDSTREAM (GETSH SPEC))
(RETURN NODENAME))
```

[127]

```
(STATE
  [NLAMBDA L
```

```
(* edited:
" 6-Aug-79 17:31")
```

```
(* STATE works like DENY and MAYBE, but gives the
```

assertion a confidence of 1.0.)

(CASSERT L 1.01)
)
(DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
(ADDTTOVAR NLAMA STATE MAYBE DENY)
(ADDTTOVAR NLAML)
(ADDTTOVAR LAMA)
)
(DECLARE: DONTCOPY
(FILEMAP (NIL (587 8708 (ASSERT 599 . 1519) (BUMP 1523 . 2340) (CASSERT
2344 . 3601) (DENY 3605 . 4054) (GETUPLE 4058 . 4347) (MATCHER 4351 .
5391) (MAYBE 5395 . 5654) (RETRIEVER 5658 . 7014) (RETVARS 7018 . 7392)
(SERT 7396 . 8408) (STATE 8412 . 8705))))))
STOP

(FILECREATED " 6-Aug-79 19:06:57" <RBECHTAL>MEMORY..17 17907

previous date: "23-Jul-79 08:04:36" <RBECHTAL>MEMORY..15)

(PRETTYCOMPRINT MEMORYCOMS)

(RPAQQ MEMORYCOMS ((VARS * (APPEND MEMORYVARS ASSERTIONS))

[IFPROP (DERIVE DERIVE* FROM FROM* NEGFROM NEGFROM* SPLIT MB MD)

*
(APPEND ASSERTIONS [MAPCAR ASSERTIONS

(FUNCTION

(LAMBDA

(X)

(GETPROP X (QUOTE FROM*)

(MAPCAR ASSERTIONS (FUNCTION

(LAMBDA (Y)

(GETPROP Y (QUOTE DERIVE*)

(P (CREATH MEMSIZE)

[MAPC ASSERTIONS (FUNCTION (LAMBDA (Q)

(SERT (EVAL Q)

Q)

(PRIN1 "Memory Reinitialized")

(TERPRI)))

(RPAQQ MEMSIZE 4374)

(RPAQQ GENNUM 10175)

```

(RPAQQ ASSERTIONS (A0175 A0174 A0173 A0172 A0171 A0170 A0169 A0168
A0167 A0166 A0165 A0164 A0163 A0162 A0161
A0160 A0159 A0158 A0157 A0156 A0155 A0154
A0153 A0152 A0151 A0150 A0149 A0148 A0147
A0146 A0145 A0144 A0143 A0142 A0141 A0140
A0139 A0138 A0137 A0136 A0135 A0134 A0133
A0132 A0131 A0130 A0129 A0128 A0127 A0126
A0125 A0124 A0123 A0122 A0121 A0120 A0119
A0118 A0117 A0116 A0115 A0114 A0113 A0112
A0111 A0110 A0109 A0108 A0107 A0106 A0105
A0104 A0103 A0102 A0101 A0100 A0099 A0098
A0097 A0096 A0095 A0094 A0093 A0092 A0091
A0090 A0089 A0088 A0087 A0086 A0085 A0084
A0083 A0082 A0081 A0080 A0079 A0078 A0077
A0076 A0075 A0074 A0073 A0072 A0071 A0070
A0069 A0068 A0067 A0066 A0065 A0064 A0063
A0062 A0061 A0060 A0059 A0058 A0057 A0056
A0055 A0054 A0053 A0052 A0051 A0050 A0049
A0048 A0047 A0046 A0045 A0044 A0043 A0042
A0041 A0040 A0039 A0038 A0037 A0036 A0035
A0034 A0033 A0032 A0031 A0030 A0029 A0028
A0027 A0026 A0025 A0024 A0023 A0022 A0021
A0020 A0019 A0018 A0017 A0016 A0015))

```

(RPAQQ MEMORYVARS (MEMSIZE GENNUM ASSERTIONS MEMORYVARS SYMBOLS))

(RPAQQ SYMBOLS NIL)

(RPAQQ A0175 (CLASS VIKING S-3A))
(RPAQQ A0174 (MEDIUM VIKING AIR))
(RPAQQ A0173 (TYPE VIKING RECONNISANCE))
(RPAQQ A0172 (ID-AMPLIFY VIKING MIL-AUXIL))
(RPAQQ A0171 (ID VIKING FRIEND))
(RPAQQ A0170 (PLATFORM VIKING))
(RPAQQ A0169 (CLASS SEASPRITE SH-2F))
(RPAQQ A0168 (MEDIUM SEASPRITE AIR))
(RPAQQ A0167 (TYPE SEASPRITE HELICOPTER))
(RPAQQ A0166 (ID-AMPLIFY SEASPRITE MIL-BATTLE))
(RPAQQ A0165 (ID SEASPRITE FRIEND))
(RPAQQ A0164 (PLATFORM SEASPRITE))
(RPAQQ A0163 (CLASS ORION P-3C))
(RPAQQ A0162 (MEDIUM ORION AIR))
(RPAQQ A0161 (TYPE ORION RECONNISANCE))
(RPAQQ A0160 (ID-AMPLIFY ORION MIL-AUXIL))
(RPAQQ A0159 (ID ORION FRIEND))
(RPAQQ A0158 (PLATFORM ORION))
(RPAQQ A0157 (CLASS HORMONE KA-25))
(RPAQQ A0156 (MEDIUM HORMONE AIR))
(RPAQQ A0155 (TYPE HORMONE HELICOPTER))
(RPAQQ A0154 (ID-AMPLIFY HORMONE MIL-BATTLE))
(RPAQQ A0153 (ID HORMONE HOSTILE))
(RPAQQ A0152 (PLATFORM HORMONE))
(RPAQQ A0151 (CLASS HAWKEYE E-2B))
(RPAQQ A0150 (MEDIUM HAWKEYE AIR))
(RPAQQ A0149 (TYPE HAWKEYE RECONNISANCE))
(RPAQQ A0148 (ID-AMPLIFY HAWKEYE MIL-AUXIL))
(RPAQQ A0147 (ID HAWKEYE FRIEND))

(RPAQQ A0146 (PLATFORM HAWKEYE))
(RPAQQ A0145 (CLASS HARRIER AV-8A))
(RPAQQ A0144 (MEDIUM HARRIER AIR))
(RPAQQ A0143 (TYPE HARRIER FIGHTER))
(RPAQQ A0142 (ID-AMPLIFY HARRIER MIL-BATTLE))
(RPAQQ A0141 (ID HARRIER FRIEND))
(RPAQQ A0140 (PLATFORM HARRIER))
(RPAQQ A0139 (CLASS FOXBAT MIG25))
(RPAQQ A0138 (MEDIUM FOXBAT AIR))
(RPAQQ A0137 (TYPE FOXBAT FIGHTER))
(RPAQQ A0136 (ID-AMPLIFY FOXBAT MIL-BATTLE))
(RPAQQ A0135 (ID FOXBAT HOSTILE))
(RPAQQ A0134 (PLATFORM FOXBAT))
(RPAQQ A0133 (CLASS CORSAIR A-7))
(RPAQQ A0132 (MEDIUM CORSAIR AIR))
(RPAQQ A0131 (TYPE CORSAIR FIGHTER))
(RPAQQ A0130 (ID-AMPLIFY CORSAIR MIL-BATTLE))
(RPAQQ A0129 (ID CORSAIR FRIEND))
(RPAQQ A0128 (PLATFORM CORSAIR))
(RPAQQ A0127 (CLASS BACKFIRE RV-G))
(RPAQQ A0126 (MEDIUM BACKFIRE AIR))
(RPAQQ A0125 (TYPE BACKFIRE BOMBER))
(RPAQQ A0124 (ID-AMPLIFY BACKFIRE MIL-BATTLE))
(RPAQQ A0123 (ID BACKFIRE HOSTILE))
(RPAQQ A0122 (PLATFORM BACKFIRE))
(RPAQQ A0121 (CLASS RATHBURNE KNOX))
(RPAQQ A0120 (MEDIUM RATHBURNE SURFACE))
(RPAQQ A0119 (TYPE RATHBURNE FRIGATE))

(RPAQQ A0118 (ID-AMPLIFY RATHBURNE MIL-BATTLE))
(RPAQQ A0117 (ID RATHBURNE FRIEND))
(RPAQQ A0116 (PLATFORM RATHBURNE))
(RPAQQ A0115 (CLASS YANK-1 YANKEE))
(RPAQQ A0114 (MEDIUM YANK-1 SUB))
(RPAQQ A0113 (TYPE YANK-1 SUB))
(RPAQQ A0112 (ID-AMPLIFY YANK-1 MIL-BATTLE))
(RPAQQ A0111 (ID YANK-1 HOSTILE))
(RPAQQ A0110 (PLATFORM YANK-1))
(RPAQQ A0109 (CLASS WAINWRIGHT BELKNAP))
(RPAQQ A0108 (MEDIUM WAINWRIGHT SURFACE))
(RPAQQ A0107 (TYPE WAINWRIGHT CRUISER))
(RPAQQ A0106 (ID-AMPLIFY WAINWRIGHT MIL-BATTLE))
(RPAQQ A0105 (ID WAINWRIGHT FRIEND))
(RPAQQ A0104 (PLATFORM WAINWRIGHT))
(RPAQQ A0103 (CLASS SUNFISH STURGEON))
(RPAQQ A0102 (MEDIUM SUNFISH SUB))
(RPAQQ A0101 (TYPE SUNFISH SUB))
(RPAQQ A0100 (ID-AMPLIFY SUNFISH MIL-BATTLE))
(RPAQQ A0099 (ID SUNFISH FRIEND))
(RPAQQ A0098 (PLATFORM SUNFISH))
(RPAQQ A0097 (CLASS PROVORNY KASHIN))
(RPAQQ A0096 (MEDIUM PROVORNY SURFACE))
(RPAQQ A0095 (TYPE PROVORNY DESTROYER))
(RPAQQ A0094 (ID-AMPLIFY PROVORNY MIL-BATTLE))
(RPAQQ A0093 (ID PROVORNY HOSTILE))
(RPAQQ A0092 (PLATFORM PROVORNY))
(RPAQQ A0091 (CLASS MINSK KIEV))
(RPAQQ A0090 (MEDIUM MINSK SURFACE))

(RPAQQ A0089 (TYPE MINSK CARRIER))
(RPAQQ A0088 (ID-AMPLIFY MINSK MIL-BATTLE))
(RPAQQ A0087 (ID MINSK HOSTILE))
(RPAQQ A0086 (PLATFORM MINSK))
(RPAQQ A0085 (CLASS MEYERCORD KNOX))
(RPAQQ A0084 (MEDIUM MEYERCORD SURFACE))
(RPAQQ A0083 (TYPE MEYERCORD FRIGATE))
(RPAQQ A0082 (ID-AMPLIFY MEYERCORD MIL-BATTLE))
(RPAQQ A0081 (ID MEYERCORD FRIEND))
(RPAQQ A0080 (PLATFORM MEYERCORD))
(RPAQQ A0079 (CLASS LAWRENCE CHAS.ADAMS))
(RPAQQ A0078 (MEDIUM LAWRENCE SURFACE))
(RPAQQ A0077 (TYPE LAWRENCE DESTROYER))
(RPAQQ A0076 (ID-AMPLIFY LAWRENCE MIL-BATTLE))
(RPAQQ A0075 (ID LAWRENCE FRIEND))
(RPAQQ A0074 (PLATFORM LAWRENCE))
(RPAQQ A0073 (CLASS HASSAYAMPA NEOSHO))
(RPAQQ A0072 (MEDIUM HASSAYAMPA SURFACE))
(RPAQQ A0071 (TYPE HASSAYAMPA OILER))
(RPAQQ A0070 (ID-AMPLIFY HASSAYAMPA MIL-AUXIL))
(RPAQQ A0069 (ID HASSAYAMPA FRIEND))
(RPAQQ A0068 (PLATFORM HASSAYAMPA))
(RPAQQ A0067 (CLASS HALSEY LEAHY))
(RPAQQ A0066 (MEDIUM HALSEY SURFACE))
(RPAQQ A0065 (TYPE HALSEY CRUISER))
(RPAQQ A0064 (ID-AMPLIFY HALSEY MIL-BATTLE))
(RPAQQ A0063 (ID HALSEY FRIEND))
(RPAQQ A0062 (PLATFORM HALSEY))

(RPAQQ A0061 (CLASS ECHO-1 ECHOII))
(RPAQQ A0060 (MEDIUM ECHO-1 SUB))
(RPAQQ A0059 (TYPE ECHO-1 SUB))
(RPAQQ A0058 (ID-AMPLIFY ECHO-1 MIL-BATTLE))
(RPAQQ A0057 (ID ECHO-1 HOSTILE))
(RPAQQ A0056 (PLATFORM ECHO-1))
(RPAQQ A0055 (CLASS DESNA KAZBEK))
(RPAQQ A0054 (MEDIUM DESNA SURFACE))
(RPAQQ A0053 (TYPE DESNA OILER))
(RPAQQ A0052 (ID-AMPLIFY DESNA MIL-AUXIL))
(RPAQQ A0051 (ID DESNA HOSTILE))
(RPAQQ A0050 (PLATFORM DESNA))
(RPAQQ A0049 (CLASS CONSTELLATION KITTYHAWK))
(RPAQQ A0048 (MEDIUM CONSTELLATION SURFACE))
(RPAQQ A0047 (TYPE CONSTELLATION CARRIER))
(RPAQQ A0046 (ID-AMPLIFY CONSTELLATION MIL-BATTLE))
(RPAQQ A0045 (ID CONSTELLATION FRIEND))
(RPAQQ A0044 (PLATFORM CONSTELLATION))
(RPAQQ A0043 (CLASS ADMIRAL% MAKAROV KRESTAI))
(RPAQQ A0042 (MEDIUM ADMIRAL% MAKAROV SURFACE))
(RPAQQ A0041 (TYPE ADMIRAL% MAKAROV CRUISER))
(RPAQQ A0040 (ID-AMPLIFY ADMIRAL% MAKAROV MIL-BATTLE))
(RPAQQ A0039 (ID ADMIRAL% MAKAROV HOSTILE))
(RPAQQ A0038 (PLATFORM ADMIRAL% MAKAROV))
(RPAQQ A0037 (CLASS ADMIRAL% GOLOVKO KYNDA))
(RPAQQ A0036 (MEDIUM ADMIRAL% GOLOVKO SURFACE))
(RPAQQ A0035 (TYPE ADMIRAL% GOLOVKO CRUISER))
(RPAQQ A0034 (ID-AMPLIFY ADMIRAL% GOLOVKO MIL-BATTLE))
(RPAQQ A0033 (ID ADMIRAL% GOLOVKO HOSTILE))

(RPAQQ A0032 (PLATFORM ADMIRAL& GOLOVKO))

(RPAQQ A0031 (CLASS CONNOLE KNOX))

(RPAQQ A0030 (TYPE CONNOLE FRIGATE))

(RPAQQ A0029 (ID-AMPLIFY CONNOLE MIL-BATTLE))

(RPAQQ A0028 (ID CONNOLE FRIEND))

(RPAQQ A0027 (OWNSHIP CONNOLE))

(RPAQQ A0026 (LOCATION LANE3 ((55.66 -39.84)
(57.23 -36.36)
(58.56 -32.89)
(59.77 -29.01)
(61.17 -23.79)
(62.08 -19.37)
(62.99 -13.96)
(63.79 -6.72))))

(RPAQQ A0025 (LOCATION LANE2 ((56.04 -42.25)
(58.45 -37.9)
(60.37 -33.75)
(61.85 -29.94)
(63.19 -26.0)
(64.01 -22.99))))

(RPAQQ A0024 (LOCATION LANE1 ((68.93 -13.82)
(68.39 -16.57)
(66.79 -23.11)
(66.11 -25.32)
(65.02 -28.53)
(64.19 -30.47)
(63.34 -32.47)
(62.11 -35.08)
(60.64 -37.76)
(59.21 -40.16)
(58.14 -41.7))))

(RPAQQ A0023 (FROM-PORT LANE3 ST.JOHNS))

(RPAQQ A0022 (TO-PORT LANE3 MURMANSK))

(RPAQQ A0021 (FROM-PORT LANE2 ST.JOHNS))

(RPAQQ A0020 (TO-PORT LANE2 REYKJAVIK))

(RPAQQ A0019 (FROM-PORT LANE1 MURMANSK))

(RPAQQ A0018 (TO-PORT LANE1 REYKJAVIK))

(RPAQQ A0017 (MERCHANLANE LANE3))

(RPAQQ A0016 (MERCHANLANE LANE2))

<RBECHTAL>MEMORY..17

Page 77

(RPAQQ A0015 (MERCHANTLANE LANE1))

(PUTPROPS A0175 MB 1.0)

(PUTPROPS A0174 MB 1.0)

(PUTPROPS A0173 MB 1.0)

(PUTPROPS A0172 MB 1.0)

(PUTPROPS A0171 MB 1.0)

(PUTPROPS A0170 MB 1.0)

(PUTPROPS A0169 MB 1.0)

(PUTPROPS A0168 MB 1.0)

(PUTPROPS A0167 MB 1.0)

(PUTPROPS A0166 MB 1.0)

(PUTPROPS A0165 MB 1.0)

(PUTPROPS A0164 MB 1.0)

(PUTPROPS A0163 MB 1.0)

(PUTPROPS A0162 MB 1.0)

(PUTPROPS A0161 MB 1.0)

(PUTPROPS A0160 MB 1.0)

(PUTPROPS A0159 MB 1.0)

(PUTPROPS A0158 MB 1.0)

(PUTPROPS A0157 MB 1.0)

(PUTPROPS A0156 MB 1.0)

(PUTPROPS A0155 MB 1.0)

(PUTPROPS A0154 MB 1.0)

(PUTPROPS A0153 MB 1.0)

(PUTPROPS A0152 MB 1.0)

(PUTPROPS A0151 MB 1.0)

(PUTPROPS A0150 MB 1.0)

(PUTPROPS A0149 MB 1.0)

(PUTPROPS A0148 MB 1.0)

(PUTPROPS A0147 MB 1.0)
(PUTPROPS A0146 MB 1.0)
(PUTPROPS A0145 MB 1.0)
(PUTPROPS A0144 MB 1.0)
(PUTPROPS A0143 MB 1.0)
(PUTPROPS A0142 MB 1.0)
(PUTPROPS A0141 MB 1.0)
(PUTPROPS A0140 MB 1.0)
(PUTPROPS A0139 MB 1.0)
(PUTPROPS A0138 MB 1.0)
(PUTPROPS A0137 MB 1.0)
(PUTPROPS A0136 MB 1.0)
(PUTPROPS A0135 MB 1.0)
(PUTPROPS A0134 MB 1.0)
(PUTPROPS A0133 MB 1.0)
(PUTPROPS A0132 MB 1.0)
(PUTPROPS A0131 MB 1.0)
(PUTPROPS A0130 MB 1.0)
(PUTPROPS A0129 MB 1.0)
(PUTPROPS A0128 MB 1.0)
(PUTPROPS A0127 MB 1.0)
(PUTPROPS A0126 MB 1.0)
(PUTPROPS A0125 MB 1.0)
(PUTPROPS A0124 MB 1.0)
(PUTPROPS A0123 MB 1.0)
(PUTPROPS A0122 MB 1.0)
(PUTPROPS A0121 MB 1.0)
(PUTPROPS A0120 MB 1.0)

(PUTPROPS A0119 MB 1.0)
(PUTPROPS A0118 MB 1.0)
(PUTPROPS A0117 MB 1.0)
(PUTPROPS A0116 MB 1.0)
(PUTPROPS A0115 MB 1.0)
(PUTPROPS A0114 MB 1.0)
(PUTPROPS A0113 MB 1.0)
(PUTPROPS A0112 MB 1.0)
(PUTPROPS A0111 MB 1.0)
(PUTPROPS A0110 MB 1.0)
(PUTPROPS A0109 MB 1.0)
(PUTPROPS A0108 MB 1.0)
(PUTPROPS A0107 MB 1.0)
(PUTPROPS A0106 MB 1.0)
(PUTPROPS A0105 MB 1.0)
(PUTPROPS A0104 MB 1.0)
(PUTPROPS A0103 MB 1.0)
(PUTPROPS A0102 MB 1.0)
(PUTPROPS A0101 MB 1.0)
(PUTPROPS A0100 MB 1.0)
(PUTPROPS A0099 MB 1.0)
(PUTPROPS A0098 MB 1.0)
(PUTPROPS A0097 MB 1.0)
(PUTPROPS A0096 MB 1.0)
(PUTPROPS A0095 MB 1.0)
(PUTPROPS A0094 MB 1.0)
(PUTPROPS A0093 MB 1.0)
(PUTPROPS A0092 MB 1.0)
(PUTPROPS A0091 MB 1.0)

(PUTPROPS A0090 MB 1.0)
(PUTPROPS A0089 MB 1.0)
(PUTPROPS A0088 MB 1.0)
(PUTPROPS A0087 MB 1.0)
(PUTPROPS A0086 MB 1.0)
(PUTPROPS A0085 MB 1.0)
(PUTPROPS A0084 MB 1.0)
(PUTPROPS A0083 MB 1.0)
(PUTPROPS A0082 MB 1.0)
(PUTPROPS A0081 MB 1.0)
(PUTPROPS A0080 MB 1.0)
(PUTPROPS A0079 MB 1.0)
(PUTPROPS A0078 MB 1.0)
(PUTPROPS A0077 MB 1.0)
(PUTPROPS A0076 MB 1.0)
(PUTPROPS A0075 MB 1.0)
(PUTPROPS A0074 MB 1.0)
(PUTPROPS A0073 MB 1.0)
(PUTPROPS A0072 MB 1.0)
(PUTPROPS A0071 MB 1.0)
(PUTPROPS A0070 MB 1.0)
(PUTPROPS A0069 MB 1.0)
(PUTPROPS A0068 MB 1.0)
(PUTPROPS A0067 MB 1.0)
(PUTPROPS A0066 MB 1.0)
(PUTPROPS A0065 MB 1.0)
(PUTPROPS A0064 MB 1.0)
(PUTPROPS A0063 MB 1.0)

(PUTPROPS A0062 MB 1.0)
(PUTPROPS A0061 MB 1.0)
(PUTPROPS A0060 MB 1.0)
(PUTPROPS A0059 MB 1.0)
(PUTPROPS A0058 MB 1.0)
(PUTPROPS A0057 MB 1.0)
(PUTPROPS A0056 MB 1.0)
(PUTPROPS A0055 MB 1.0)
(PUTPROPS A0054 MB 1.0)
(PUTPROPS A0053 MB 1.0)
(PUTPROPS A0052 MB 1.0)
(PUTPROPS A0051 MB 1.0)
(PUTPROPS A0050 MB 1.0)
(PUTPROPS A0049 MB 1.0)
(PUTPROPS A0048 MB 1.0)
(PUTPROPS A0047 MB 1.0)
(PUTPROPS A0046 MB 1.0)
(PUTPROPS A0045 MB 1.0)
(PUTPROPS A0044 MB 1.0)
(PUTPROPS A0043 MB 1.0)
(PUTPROPS A0042 MB 1.0)
(PUTPROPS A0041 MB 1.0)
(PUTPROPS A0040 MB 1.0)
(PUTPROPS A0039 MB 1.0)
(PUTPROPS A0038 MB 1.0)
(PUTPROPS A0037 MB 1.0)
(PUTPROPS A0036 MB 1.0)
(PUTPROPS A0035 MB 1.0)
(PUTPROPS A0034 MB 1.0)

(PUTPROPS A0033 MB 1.0)
(PUTPROPS A0032 MB 1.0)
(PUTPROPS A0031 MB 1.0)
(PUTPROPS A0030 MB 1.0)
(PUTPROPS A0029 MB 1.0)
(PUTPROPS A0028 MB 1.0)
(PUTPROPS A0027 MB 1.0)
(PUTPROPS A0026 MB 1.0)
(PUTPROPS A0025 MB 1.0)
(PUTPROPS A0024 MB 1.0)
(PUTPROPS A0023 MB 1.0)
(PUTPROPS A0022 MB 1.0)
(PUTPROPS A0021 MB 1.0)
(PUTPROPS A0020 MB 1.0)
(PUTPROPS A0019 MB 1.0)
(PUTPROPS A0018 MB 1.0)
(PUTPROPS A0017 MB 1.0)
(PUTPROPS A0016 MB 1.0)
(PUTPROPS A0015 MB 1.0)
(PUTPROPS A0175 MD 0.0)
(PUTPROPS A0174 MD 0.0)
(PUTPROPS A0173 MD 0.0)
(PUTPROPS A0172 MD 0.0)
(PUTPROPS A0171 MD 0.0)
(PUTPROPS A0170 MD 0.0)
(PUTPROPS A0169 MD 0.0)
(PUTPROPS A0168 MD 0.0)
(PUTPROPS A0167 MD 0.0)

(PUTPROPS A0166 MD 0.0)
(PUTPROPS A0165 MD 0.0)
(PUTPROPS A0164 MD 0.0)
(PUTPROPS A0163 MD 0.0)
(PUTPROPS A0162 MD 0.0)
(PUTPROPS A0161 MD 0.0)
(PUTPROPS A0160 MD 0.0)
(PUTPROPS A0159 MD 0.0)
(PUTPROPS A0158 MD 0.0)
(PUTPROPS A0157 MD 0.0)
(PUTPROPS A0156 MD 0.0)
(PUTPROPS A0155 MD 0.0)
(PUTPROPS A0154 MD 0.0)
(PUTPROPS A0153 MD 0.0)
(PUTPROPS A0152 MD 0.0)
(PUTPROPS A0151 MD 0.0)
(PUTPROPS A0150 MD 0.0)
(PUTPROPS A0149 MD 0.0)
(PUTPROPS A0148 MD 0.0)
(PUTPROPS A0147 MD 0.0)
(PUTPROPS A0146 MD 0.0)
(PUTPROPS A0145 MD 0.0)
(PUTPROPS A0144 MD 0.0)
(PUTPROPS A0143 MD 0.0)
(PUTPROPS A0142 MD 0.0)
(PUTPROPS A0141 MD 0.0)
(PUTPROPS A0140 MD 0.0)
(PUTPROPS A0139 MD 0.0)
(PUTPROPS A0138 MD 0.0)

(PUTPROPS A0137 MD 0.0)
(PUTPROPS A0136 MD 0.0)
(PUTPROPS A0135 MD 0.0)
(PUTPROPS A0134 MD 0.0)
(PUTPROPS A0133 MD 0.0)
(PUTPROPS A0132 MD 0.0)
(PUTPROPS A0131 MD 0.0)
(PUTPROPS A0130 MD 0.0)
(PUTPROPS A0129 MD 0.0)
(PUTPROPS A0128 MD 0.0)
(PUTPROPS A0127 MD 0.0)
(PUTPROPS A0126 MD 0.0)
(PUTPROPS A0125 MD 0.0)
(PUTPROPS A0124 MD 0.0)
(PUTPROPS A0123 MD 0.0)
(PUTPROPS A0122 MD 0.0)
(PUTPROPS A0121 MD 0.0)
(PUTPROPS A0120 MD 0.0)
(PUTPROPS A0119 MD 0.0)
(PUTPROPS A0118 MD 0.0)
(PUTPROPS A0117 MD 0.0)
(PUTPROPS A0116 MD 0.0)
(PUTPROPS A0115 MD 0.0)
(PUTPROPS A0114 MD 0.0)
(PUTPROPS A0113 MD 0.0)
(PUTPROPS A0112 MD 0.0)
(PUTPROPS A0111 MD 0.0)
(PUTPROPS A0110 MD 0.0)

(PUTPROPS A0109 MD 0.0)
(PUTPROPS A0108 MD 0.0)
(PUTPROPS A0107 MD 0.0)
(PUTPROPS A0106 MD 0.0)
(PUTPROPS A0105 MD 0.0)
(PUTPROPS A0104 MD 0.0)
(PUTPROPS A0103 MD 0.0)
(PUTPROPS A0102 MD 0.0)
(PUTPROPS A0101 MD 0.0)
(PUTPROPS A0100 MD 0.0)
(PUTPROPS A0099 MD 0.0)
(PUTPROPS A0098 MD 0.0)
(PUTPROPS A0097 MD 0.0)
(PUTPROPS A0096 MD 0.0)
(PUTPROPS A0095 MD 0.0)
(PUTPROPS A0094 MD 0.0)
(PUTPROPS A0093 MD 0.0)
(PUTPROPS A0092 MD 0.0)
(PUTPROPS A0091 MD 0.0)
(PUTPROPS A0090 MD 0.0)
(PUTPROPS A0089 MD 0.0)
(PUTPROPS A0088 MD 0.0)
(PUTPROPS A0087 MD 0.0)
(PUTPROPS A0086 MD 0.0)
(PUTPROPS A0085 MD 0.0)
(PUTPROPS A0084 MD 0.0)
(PUTPROPS A0083 MD 0.0)
(PUTPROPS A0082 MD 0.0)
(PUTPROPS A0081 MD 0.0)

(PUTPROPS A0080 MD 0.0)
(PUTPROPS A0079 MD 0.0)
(PUTPROPS A0078 MD 0.0)
(PUTPROPS A0077 MD 0.0)
(PUTPROPS A0076 MD 0.0)
(PUTPROPS A0075 MD 0.0)
(PUTPROPS A0074 MD 0.0)
(PUTPROPS A0073 MD 0.0)
(PUTPROPS A0072 MD 0.0)
(PUTPROPS A0071 MD 0.0)
(PUTPROPS A0070 MD 0.0)
(PUTPROPS A0069 MD 0.0)
(PUTPROPS A0068 MD 0.0)
(PUTPROPS A0067 MD 0.0)
(PUTPROPS A0066 MD 0.0)
(PUTPROPS A0065 MD 0.0)
(PUTPROPS A0064 MD 0.0)
(PUTPROPS A0063 MD 0.0)
(PUTPROPS A0062 MD 0.0)
(PUTPROPS A0061 MD 0.0)
(PUTPROPS A0060 MD 0.0)
(PUTPROPS A0059 MD 0.0)
(PUTPROPS A0058 MD 0.0)
(PUTPROPS A0057 MD 0.0)
(PUTPROPS A0056 MD 0.0)
(PUTPROPS A0055 MD 0.0)
(PUTPROPS A0054 MD 0.0)
(PUTPROPS A0053 MD 0.0)

(PUTPROPS A0052 MD 0.0)
(PUTPROPS A0051 MD 0.0)
(PUTPROPS A0050 MD 0.0)
(PUTPROPS A0049 MD 0.0)
(PUTPROPS A0048 MD 0.0)
(PUTPROPS A0047 MD 0.0)
(PUTPROPS A0046 MD 0.0)
(PUTPROPS A0045 MD 0.0)
(PUTPROPS A0044 MD 0.0)
(PUTPROPS A0043 MD 0.0)
(PUTPROPS A0042 MD 0.0)
(PUTPROPS A0041 MD 0.0)
(PUTPROPS A0040 MD 0.0)
(PUTPROPS A0039 MD 0.0)
(PUTPROPS A0038 MD 0.0)
(PUTPROPS A0037 MD 0.0)
(PUTPROPS A0036 MD 0.0)
(PUTPROPS A0035 MD 0.0)
(PUTPROPS A0034 MD 0.0)
(PUTPROPS A0033 MD 0.0)
(PUTPROPS A0032 MD 0.0)
(PUTPROPS A0031 MD 0.0)
(PUTPROPS A0030 MD 0.0)
(PUTPROPS A0029 MD 0.0)
(PUTPROPS A0028 MD 0.0)
(PUTPROPS A0027 MD 0.0)
(PUTPROPS A0026 MD 0.0)
(PUTPROPS A0025 MD 0.0)
(PUTPROPS A0024 MD 0.0)

(PUTPROPS A0023 MD 0.0)

(PUTPROPS A0022 MD 0.0)

(PUTPROPS A0021 MD 0.0)

(PUTPROPS A0020 MD 0.0)

(PUTPROPS A0019 MD 0.0)

(PUTPROPS A0018 MD 0.0)

(PUTPROPS A0017 MD 0.0)

(PUTPROPS A0016 MD 0.0)

(PUTPROPS A0015 MD 0.0)

(CREATH MEMSIZE)

[MAPC ASSERTIONS (FUNCTION (LAMBDA (Q)
 (SERT (EVAL Q)
 Q])

(PRIN1 "Memory Reinitialized")

(TERPRI)

(DECLARE: DONTCOPY

 (FILEMAP (NIL)))

STOP

(FILECREATED "23-Aug-79 17:56:55" <RBECHTAL>MSGMTR..27 16895

changes to: WEATHERMSG

previous date: " 6-Aug-79 09:36:54" <RBECHTAL>MSGMTR..26)

(PRETTYCOMPRINT MSGMTRCOMS)

(RPAQQ MSGMTRCOMS ((VARS * MSGMTRVARS)
(FNS * MSGMTRFNS)))

(RPAQQ MSGMTRVARS (DSPLAYFLG MSGFILE OWNSHIP SENSORANGE CURTIME))

(RPAQQ DSPLAYFLG NIL)

(RPAQQ MSGFILE SCENE.ICE)

(RPAQQ OWNSHIP CONNOLE)

(RPAQQ SENSORANGE 25)

(RPAQQ CURTIME 0)

(RPAQQ MSGMTRFNS (BEYONDINTEREST DESCRIBMSG DISPCHECK DISPLAY DISPLOB
DISPMARK EWMSG GREATESTPROB IDENT
INTERPOLABLE MEDIUM MELD MIDP MSGMTR
NEWSYM OWNMSG OWNPOS SENSORMSG
TWO-PLACE WEATHERMSG))

(DEFINEQ

[128]

(BEYONDINTEREST
[LAMBDA (TXT)(* edited:
"31-Jul-79 09:21")

NIL))

[129]

(DESCRIBMSG
[LAMBDA (TXT)(* edited:
" 6-Aug-79 08:53")

(* DESCRIBMSG prints the information contained in a message in a relatively nice format for the user. The function itself is fairly simple, if tedious. After determining the type of message, the information is printed. Messages concerning the home ship are ignored. After printing, if display is enabled, a picture containing the new location is drawn.)

(PROG ((WKNM (CAR TXT))
(SOURCE (CADR TXT))

AD-A084 053

SDC INTEGRATED SERVICES INC SAN DIEGO CA

F/6 9/2

STAMMER2 PRODUCTION SYSTEM FOR TACTICAL SITUATION ASSESSMENT. V--ETC(U)

OCT 79 D C MCCALL, P H MORRIS, D F KIBLER

N00123-76-C-0172

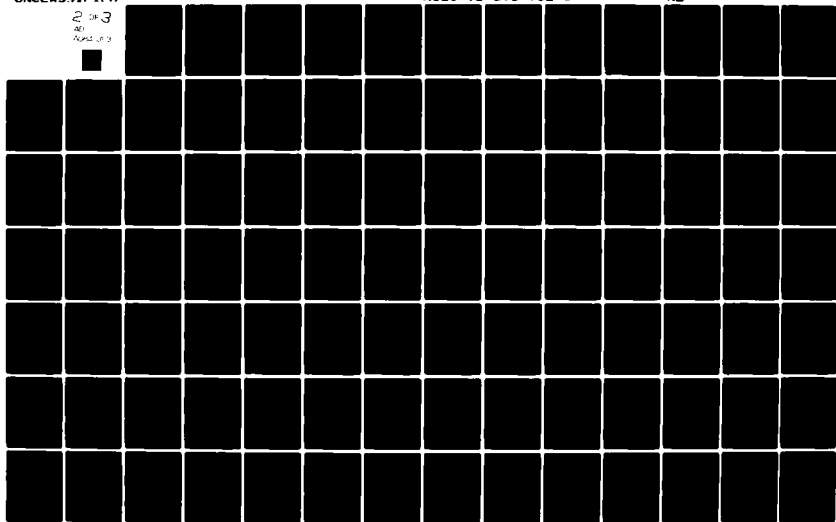
UNCLASSIFIED

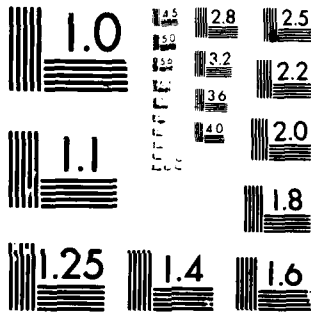
NOSC-TD-298-VOL-2

NL

2 OF 3

AD-A084 053





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

    (TIME (CAR (LAST TXT)))
    POS)
  (SETQ CURTIME TIME)
  (COND
    ((EQ WKNM OWNSHIP)
     (RETURN)))
  [COND
    ((EQ (CADR TXT)
     (QUOTE WEATHER))
     (TERPRI)
     (SPACES 5)
     (PRIN1 "Weather report.")
     (TERPRI)
     (SPACES 5)
     (PRIN1 "Storm centered at ")
     (PRIN1 (CENTROID (CADDR TXT)))
     (TERPRI)
     (COND
       (DSPLAYFLG (TERPRI)
        (SPACES 10)
        (PRIN1 "Display follows")
        (TERPRI)
        (WAITER)
        (DSPCMD "PTR PL,ST"))
      (RETURN))
    [(NUMBERP (CADDR TXT))
     (SETQ POS (LIST (CADDR TXT)
                     (CADDRD TXT)))
     (COND
       ((OR (EQ SOURCE (QUOTE RADAR))
            (EQ SOURCE (QUOTE SONAR)))
        (TERPRI)
        (SPACES 5)
        (PRIN1 SOURCE)
        (PRIN1 " contact at ")
        (PRIN1 POS)
        (PRIN1 " Time: ")
        (PRIN1 TIME))
       (T (TERPRI)
          (SPACES 5)
          (PRIN1 "Message received from external source.")
          (TERPRI)
          (SPACES 5)
          (PRIN1 "Something detected at ")
          (PRIN1 POS)
          (PRIN1 " Time: ")
          (PRIN1 TIME]
      (T (SETQ POS (CADDRD TXT))
         (COND
           ((EQ SOURCE (QUOTE EXTERNAL))
            (TERPRI)
            (SPACES 5)
            (PRIN1
              "Message from external source. Heard something at bearing ")
            (PRIN1 POS)
            [SETQ POS (LIST (CAR (CADDRD TXT))
                          (CADR (CADDRD TXT))

```

```

      (TERPRI)
      (SPACES 5)
      (PRIN1 "Detector located at ")
      (PRIN1 POS)
      (PRIN1 " Time: ")
      (PRIN1 TIME))
    (T (TERPRI)
      (SPACES 5)
      (PRIN1 "Passive detection. Heard ")
      (PRIN1 (CADDR TXT))
      (PRIN1 " at bearing ")
      (PRIN1 POS)
      (PRIN1 " Time: ")
      (PRIN1 TIME])
  (TERPRI)
  (SPACES 5)
  (PRIN1 "Associated with track ")
  (PRIN1 WKNM)
  (TERPRI)
  (TERPRI)
  (COND
    (DSPLAYFLG (PRIN1 "      Display follows.")
      (TERPRI)
      (WAITER)
      (DPCMD "PTR PL"))

```

[130]

```

(DISPCHECK
  [LAMBDA (NAME)

```

```

      (* edited:
      " 6-Aug-79 08:55")

```

```

      (* This function checks the property list of a
      platform name to determine if the platform has
      already been placed in the display.
      If not, a DSPADDTRH needs to be performed.)

```

```

      (GETPROP NAME (QUOTE INDISPLAY))

```

[131]

```

(DISPLAY
  [LAMBDA (PLATNAME LAT LON TIME)

```

```

      (* edited:
      " 6-Aug-79 08:56")

```

```

      (* DISPLAY does the necessary stuff to get a new
      platform sighting added to the display file.
      If the platform is new to the display, it is added
      to the display file, otherwise, only the new
      sighting is added.)

```

```

      (COND
        (DSPLAYFLG (COND
          ((DISPCHECK PLATNAME)
            (DSPADDINC PLATNAME LAT LON (FLOAT TIME)))

```

```

(T (DSPADDTRH PLATNAME (QUOTE PL)
    (MELD (IDENT PLATNAME)
        (MEDIUM PLATNAME)))
  (DISPMARK PLATNAME)
  (DSPADDINC PLATNAME LAT LON (FLOAT TIME)))

```

[132]

```

(DISPLOB
 [LAMBDA (PNAME SPOS DPOS TIME)

```

```

(* edited:
 " 6-Aug-79 08:59")

```

```

(* DISPLOB displays lines of bearing
 (such as those obtained by EW sightings) by adding
 the midpoint of a line drawn from the detecting
 craft location to a point maxsensorange miles
 (50) along the given bearing to the display.)

```

```

(PROG (TEMP1 TEMP2)
 (COND

```

```

  (DSPLAYFLG (SETQ TEMP1 (MIDP (CAR SPOS)
                                (CAR DPOS)))
              (SETQ TEMP2 (MIDP (CADR SPOS)
                                (CADR DPOS)))
  (DISPLAY PNAME TEMP1 TEMP2 TIME))

```

[133]

```

(DISPMARK
 [LAMBDA (NAME)

```

```

(* edited:
 " 6-Aug-79 09:00")

```

```

(* DISPMARK places a marker on the property list of
 platforms that have been entered into the display
 file.)

```

```

(PUTPROP NAME (QUOTE INDISPLAY)
 T))

```

[134]

```

(EWMSG
 [LAMBDA (TXT EXTFLG)

```

```

(* edited:
 " 6-Aug-79 09:04")

```

```

(* EWMSG adds information contained in EW messages
 to the network. The information includes position
 information (a line from the detecting platform
 along the LOB for 50 miles), the time of the
 detection, and, if the detection was made by the
 homeship, the emitter detected.)

```

```

(PROG ((SNODE (NEWSYM (QUOTE SIGHTING)))
      (WKNM (CAR TXT)))

```

```

(SOURCE (CADR TXT))
(BEAR (CADDR TXT))
(EMIT (CADDRR TXT))
TEMPLACE1 TEMPLACE2 TIME)
(CASSERT (LIST (QUOTE SOURCE)
                SNODE SOURCE)
  1.0)
(COND
  (EXTFLG (SETQ TIME (CADDRR (CADDRR TXT)))
    (CASSERT
      (LIST (QUOTE POSITION)
            SNODE
            (LIST (SETQ TEMPLACE1
                      (LIST (CADDR (CDDR TXT))
                            (CADDR (CADDRR TXT))
                      (SETQ TEMPLACE2
                            (GETPOINT TEMPLACE1 BEAR SENSORANGE)
                      1.0)
            (CASSERT (LIST (QUOTE TOS)
                          SNODE TIME)
              1.0))
      (T (SETQ TIME (CADR (CADDRR TXT)))
        (CASSERT (LIST (QUOTE TOS)
                      SNODE TIME)
          1.0)
        (CASSERT (LIST (QUOTE EMITTER)
                      SNODE EMIT)
          1.0)
        (CASSERT (LIST (QUOTE POSITION)
                      SNODE
                      (LIST (SETQ TEMPLACE1 (OWNPOS TIME))
                            (SETQ TEMPLACE2
                              (GETPOINT TEMPLACE1 BEAR
                                SENSORANGE)
                              1.0)))
          1.0)))
    (CASSERT (LIST (QUOTE SIGHTING)
                  WKNM SNODE)
      1.0)
    (DISPLOB WKNM TEMPLACE1 TEMPLACE2 TIME))

```

[135]

```

(GREATESTPROB
  (LAMBDA (POSLIST)

```

```

(* edited:
  " 6-Aug-79 09:07")

```

```

(* GREATESTPROB takes a list of answers of the form
that RETRIEVER returns, and examines them, returning
the element whose confidence is highest.
If no element has positive confidence, or if more
than one element is equally likely
(at greatest confidence), GREATESTPROB returns NIL.)

```

```

(PROG (ANS (ANSCON 0.0))
  (MAPC POSLIST (FUNCTION (LAMBDA (A)
    (COND

```

```

      ((GREATERP (GETCON (CAR A))
                  ANSCON)
       (SETQ ANSCON (GETCON (CAR A)))
       (SETQ ANS A))
      ((EQP ANSCON (GETCON (CAR A)))
       (SETQ ANS NIL)
      (RETURN ANS))

```

[136]

```

(IDENT
 [LAMBDA (NAME)

```

```

(* edited:
 " 6-Aug-79 09:09")

```

```

(* Used in display initialization of platforms, this
 attempts to determine whether the platform is
 FRIEND, HOSTILE, or UNKNOWN, and returns the
 appropriate. Default is UNKNOWN.)

```

```

(PROG (POSIB ANS)
 [SETQ POSIB (RETRIEVER (LIST (QUOTE ID)
                               NAME
                               (QUOTE *WHAID]
 (SETQ ANS (GREATESTPROB POSIB))
 (COND
  (ANS (RETURN (CDDR ANS)))
  (T (RETURN (QUOTE UNKNOWN])

```

[137]

```

(INTERPOLABLE
 [LAMBDA (TXT)

```

```

(* edited:
 "31-Jul-79 09:21")

```

```

  NIL))

```

[138]

```

(MEDIUM
 [LAMBDA (NAME)

```

```

(* edited:
 " 6-Aug-79 09:11")

```

```

(* Determines the medium of a platform.
 Default is SURFACE (as opposed to SUB or AIR).)

```

```

(PROG (TEMP1 RETURNER)
 [SETQ TEMP1 (RETRIEVER (LIST (QUOTE MEDIUM)
                               NAME
                               (QUOTE **WHAMED]
 (SETQ RETURNER (GREATESTPROB TEMP1))
 (COND
  (RETURNER (RETURN (CDDR RETURNER)))
  (T (RETURN (QUOTE SURFACE])

```

[139]

(MELD
[LAMBDA (ID MED)

(* edited:
" 6-Aug-79 09:12")

(* Creates a DSPLA type to be used for platforms.
The type controls the symbol used in the display for
a platform.)

(PROG (A B)
 (SELECTQ ID
 (UNKNOWN (SETQ A (QUOTE U)))
 (FRIEND (SETQ A (QUOTE F)))
 (HOSTILE (SETQ A (QUOTE H)))
 (SETQ A (QUOTE U)))
 (SELECTQ MED
 (AIR (SETQ B (QUOTE A)))
 (SURFACE (SETQ B (QUOTE S)))
 (SUB (SETQ B (QUOTE U)))
 (SETQ B (QUOTE S)))
 (RETURN (PACK (LIST A B))

[140]

(MIDP
[LAMBDA (P1 P2)

(* edited:
" 6-Aug-79 09:15")

(* Returns the "average" of two latitudes or
longitudes. The 180 degree check is to insure that
the shortest distance is taken when changing sign,
especially for longitude.)

(PROG (TEMP2)
 (SETQ TEMP2 (FDIFFERENCE P1 P2))
 (COND
 [(GREATERP (ABS TEMP2)
 180.0)
 (RETURN (MINUS (TWO-PLACE (FQUOTIENT TEMP2 2.0)
 (T (RETURN (TWO-PLACE (FQUOTIENT (FPLUS P1 P2)
 2.0))

[141]

(MSGMTR
[LAMBDA NIL

(* edited:
" 6-Aug-79 09:19")

(* MSGMTR reads a message (LISP S-expression) from
the designated message file, freezes deduction,
passes the message to the appropriate handler, then
unfreezes the deductions. Returns IGNORE in those
cases where the message should have no effect on
output, returns NIL if there are no more messages,

and returns T otherwise. Messages about the home
ship are ignored for printout.
Messages out of range, or predictable from existing
information are ignored.)

```
(PROG (OLDIN MSG)
  (SETQ OLDIN (INPUT))
  (INFILE MSGFILE)
  (SETQ MSG (READ))
  (INFILE OLDIN)
  (FREEZE)
  [COND
    ((EQ MSG (QUOTE STOP))
      (CLOSEF? MSGFILE)
      (UNFREEZE)
      (RETURN))
    ((EQ (CAR MSG)
      OWNSHIP)
      (OWNMSG MSG)
      (UNFREEZE)
      (RETURN (QUOTE IGNORE)))
    ((BEYONDINTEREST MSG)
      (DESCRIBMSG MSG)
      (PRIN1 "Beyond area of interest. Ignored.")
      (TERPRI)
      (UNFREEZE)
      (RETURN (QUOTE IGNORE)))
    ((INTERPOLABLE MSG)
      (DESCRIBMSG MSG)
      (PRIN1 "Predictable from existing information. Ignored.")
      (TERPRI)
      (UNFREEZE)
      (RETURN (QUOTE IGNORE)))
    ((EQ (CADR MSG)
      (QUOTE WEATHER))
      (WEATHERMSG MSG))
    ((NUMBERP (CADDR MSG))
      (SENSORMSG MSG))
    (T (COND
      ((EQ (CADR MSG)
        (QUOTE EW))
        (EWMSG MSG))
      (T (EWMSG MSG T)
        (UNFREEZE)
        (DESCRIBMSG MSG)
        (RETURN T))
```

[142]

```
(NEWSYM
  [LAMBDA (NAME)
```

```
(* edited:
  " 6-Aug-79 09:21")
```

(* NEWSYM is a method of generating custom atoms.
It acts like GENSYM, only maintains a separate
counter for each atom, permits atoms of arbitrary

length, and has no leading zeroes in the numeric part. It is also less efficient, both in time and space.)

```
[COND
  [(GETPROP NAME (QUOTE COUNTER))
   (PUTPROP NAME (QUOTE COUNTER)
    (ADD1 (GETPROP NAME (QUOTE COUNTER))
     (T (PUTPROP NAME (QUOTE COUNTER)
      1)
      (SETQ SYMBOLS (CONS NAME SYMBOLS)
       (PACK (APPEND (UNPACK NAME)
        (UNPACK (GETPROP NAME (QUOTE COUNTER))
```

[143]

```
(OWNMSG
  [LAMBDA (TXT)
```

```
(* edited:
  " 6-Aug-79 09:23")
```

(* OWNMSG updates the location of the home ship in the data base. Time and location are added.)

```
(PROG [(SNODE (NEWSYM (QUOTE SIGHTING)
  (CASSERT (LIST (QUOTE TOS)
    SNODE
    (CADDR TXT))
    1.0)
  (CASSERT [LIST (QUOTE POSITION)
    SNODE
    (LIST (LIST (CADR TXT)
      (CADDR TXT)
      1.0)
  (CASSERT (LIST (QUOTE SIGHTING)
    OWNSHIP SNODE)
    1.0)
  (DISPLAY OWNSHIP (CADR TXT)
    (CADDR TXT)
    (CADDRR TXT))
```

[144]

```
(OWNPOS
  [LAMBDA (TIME)
```

```
(* edited:
  " 6-Aug-79 09:23")
(* OWNPOS returns the
location of the home
ship at a given time)
```

```
(CAR (PLATPOS OWNSHIP TIME))
```

[145]

```
(SENSORMSG
  [LAMBDA (TXT)
```

```
(* edited:
  " 6-Aug-79 09:25")
```

(* SENSORMSG adds information from sonar and radar sightings to the data base. Sonar and radar messages include time and location information. The source of the message is also added. If there is an indication of video (radar) or return (sonar) strength, this is also added.)

```
(PROG [(SNODE (NEWSYM (QUOTE SIGHTING)))
      (WKNM (CAR TXT))
      (SOURCE (CADR TXT))
      (LAT (CADDR TXT))
      (STR (CAR (CDDDDR TXT)))
      (LON (CDDDDR TXT))
      (TIME (CAR (LAST TXT)
(CASSERT (LIST (QUOTE POSITION)
                SNODE
                (LIST (LIST LAT LON)))
          1.0)
(CASSERT (LIST (QUOTE SOURCE)
                SNODE SOURCE)
          1.0)
(CASSERT (LIST (QUOTE TOS)
                SNODE TIME)
          1.0)
(COND
  ((NOT (EQUAL STR TIME))
   (CASSERT (LIST (QUOTE STRENGTH)
                  SNODE STR)
             1.0)))
(CASSERT (LIST (QUOTE SIGHTING)
                WKNM SNODE)
          1.0)
(DISPLAY WKNM LAT LON TIME])
```

[146]

(TWO-PLACE
[LAMBDA (X)

(* edited:
" 6-Aug-79 09:26")

(* TWO-PLACE takes a number as argument, and returns that number rounded to two decimal places.)

```
(FQUOTIENT (FIX (FPLUS .5 (FTIMES X 100.0)))
            100.0])
```

[147]

(WEATHERMSG
[LAMBDA (TXT)

(* edited:
"23-Aug-79 17:56")

(* WEATHERMSG adds the information in a weather

report to memory. The information includes the polygon that defines the location of the storm and the time that the storm was sighted. If appropriate, the location of the storm is added to the display.)

```
(PROG ((SNAME (CAR TXT))
      (LOC (CADDR TXT))
      (TM (CADDRR TXT)))
  (CASSERT (LIST (QUOTE LOCATION)
                SNAME LOC)
    1.0)
  (CASSERT (LIST (QUOTE STORM)
                SNAME)
    1.0)
  (COND
    (DISPLAYFLG (DSPADDTRH SNAME (QUOTE ST)
                          (QUOTE XX))
      (MAPC LOC (FUNCTION (LAMBDA (STVER)
                          (DSPADDINC SNAME (CAR STVER)
                                      (CADR STVER)
                                      (FLOAT TM))
                        )
    )
  )
  (DECLARE: DONTCOPY
    (FILEMAP (NIL (688 16871 (BEYONDINTEREST 700 . 813) (DESCRIBMSG 817 .
3489) (DISPCHECK 3493 . 3926) (DISPLAY 3930 . 4671) (DISPLOB 4675 . 5350
) (DISPMARK 5354 . 5720) (EWMSG 5724 . 7356) (GREATESTPROB 7360 . 8173)
(IDENT 8177 . 8815) (INTERPOLABLE 8819 . 8930) (MEDIUM 8934 . 9480) (
MELD 9484 . 10157) (MIDP 10161 . 10805) (MSGMTR 10809 . 12682) (NEWSYM
12686 . 13472) (OWNMSG 13476 . 14150) (OWNPOS 14154 . 14426) (SENSORMSG
14430 . 15635) (TWO-PLACE 15639 . 15994) (WEATHERMSG 15998 . 16868))))
  STOP
```

(FILECREATED "28-Aug-79 11:42:07" <DKIBLER>NEWEXP.LSP.34 33327

changes to: NEWEXPVARS

previous date: "28-Aug-79 11:35:20" <DKIBLER>NEWEXP.LSP.33)

(PRETTYCOMPRINT NEWEXPCOMS)

(RPAQQ NEWEXPCOMS [(VARS * NEWEXPVARS)
(IFPROP PRINFORMS * RELATIONS)
(IFPROP QHPRODS * STATES)
(FNS * NEWEXPFNS)
(P (LOAD (QUOTE QH.COM))

(RPAQQ NEWEXPVARS [(ASSERTION NIL)
RELATIONS
(RULE NIL)
carriagereturn
(SMALLNUMB (QUOTE (1 2 3 4 5 6 7 8 9)))
STATES EXPLAINFLAG
(DULLREL (QUOTE (NOT-FIRST NOT-LAST CONTACT SIGHTING
INSIDE-A-MERCHANTLANE LESS-THAN
GREATER-THAN PLATFORM SAME-AS
FIRST-SIGHTING LAST-SIGHTING))

(RPAQ ASSERTION NIL)

(RPAQQ RELATIONS (CLASS OWNSHIP PLATFORM CONTACT SIGHTING SOURCE TOS
POSITION TYPE EMITTER DETECTION FIRST-SIGHTING
RADAR-MODE RANGE LESS-THAN STRENGTH MODE
GREATER-THAN SPEED LAND-DIST
REACHABLE-BY-A-COMBATANT MEDIUM INSIDE
INSIDE-A-MERCHANTLANE MERCHANTLANE IN-LANE
SUCCESSOR COURSE ROUGHLY-THE-SAME-COURSE-AS
ROUGHLY-THE-SAME-SPEED-AS ID ID-AMPLIFY
LOCATION TO-PORT FROM-PORT SAME-AS PATROL
POSSIBLE-REPORT CROSSPATHS GRAZE WENT-BEFORE
WENT-AFTER BLOCKED-FROM DISSIMILAR SWR
SIMPLY-WITHIN-REACH WITHIN-REACH NOT-FIRST
NOT-LAST ALIAS COURSEFROM SPEEDFROM))

(RPAQ RULE NIL)

(RPAQQ carriagereturn %
)

(RPAQQ SMALLNUMB (1 2 3 4 5 6 7 8 9))

(RPAQQ STATES (<EXPLTREE> <PLATIS> <VALIS> <ATTIS> <TYPIS> <IDIS>
<IDAMPIS> <WHATFORM> <WHOSE2FORM> <WHOSEFORM> <TELLABT>
<WHEREFORM> <WHEREITEM> <WHAT2FORM> <TYPE2> <ID2> <IDAMP2>
<OCCURNUM> <OTHER2>))

(RPAQQ EXPLAINFLAG NIL)

```
(RPAQQ DULLREL (NOT-FIRST NOT-LAST CONTACT SIGHTING
                INSIDE-A-MERCHANTLANE LESS-THAN GREATER-THAN |
                PLATFORM SAME-AS FIRST-SIGHTING LAST-SIGHTING)
)
```

```
(RPAQQ RELATIONS (CLASS OWNERSHIP PLATFORM CONTACT SIGHTING SOURCE TOS
                  POSITION TYPE EMITTER DETECTION FIRST-SIGHTING
                  RADAR-MODE RANGE LESS-THAN STRENGTH MODE
                  GREATER-THAN SPEED LAND-DIST
                  REACHABLE-BY-A-COMBATANT MEDIUM INSIDE
                  INSIDE-A-MERCHANTLANE MERCHANTLANE IN-LANE
                  SUCCESSOR COURSE ROUGHLY-THE-SAME-COURSE-AS
                  ROUGHLY-THE-SAME-SPEED-AS ID ID-AMPLIFY
                  LOCATION TO-PORT FROM-PORT SAME-AS PATROL
                  POSSIBLE-REPORT CROSSPATHS GRAZE WENT-BEFORE
                  WENT-AFTER BLOCKED-FROM DISSIMILAR SWR
                  SIMPLY-WITHIN-REACH WITHIN-REACH NOT-FIRST
                  NOT-LAST ALIAS COURSEFROM SPEEDFROM))
```

```
(PUTPROPS CLASS PRINFORMS ((2 " is " (MODIFIER)
                              "a " 3 T)))
```

```
(PUTPROPS OWNERSHIP PRINFORMS ((2 " is " (MODIFIER)
                                   "the OWNERSHIP" T)))
```

```
(PUTPROPS PLATFORM PRINFORMS ((2 " is " (MODIFIER)
                                   "a platform" T)
                                ("the platform " 2)))
```

```
(PUTPROPS CONTACT PRINFORMS ((2 " is " (MODIFIER)
                                   "a contact" T)
                                ("the contact " 2)))
```

```
(PUTPROPS SIGHTING PRINFORMS ((3 " is " (MODIFIER)
                                   "a sighting of " 2 T)
                                (3 " is " (MODIFIER)
                                   "a sighting of "
                                   (" of ")))
```

```
(PUTPROPS SOURCE PRINFORMS ((3 " is " (MODIFIER)
                                   "the source of " 2 T)
                                ("The source of " 2 " is " (MODIFIER)
                                   3)))
```

```
(PUTPROPS TOS PRINFORMS ((2 " occurred at " 3 T)
                          ("The time of " 2 " is " (MODIFIER)
                             3)))
```

```
(PUTPROPS POSITION PRINFORMS ((3 " is " (MODIFIER)
                                   "the position of " 2 T)
                                ("The position of " 2 " is " (MODIFIER)
                                   3 T)))
```

```
(PUTPROPS TYPE PRINFORMS ((2 " is " (MODIFIER)
                              "a " 3 T)))
```

```
(PUTPROPS EMITTER PRINFORMS ((3 " is " (MODIFIER)
```

[illegible]

"inside a merchantlane"
T)))

(PUTPROPS MERCHANTLANE PRINFORMS ((2 " is " (MODIFIER)
"a merchant lane" T)
(" the merchantlane " 2 T)))

(PUTPROPS IN-LANE PRINFORMS ((3 " is " (MODIFIER)
"in the merchantlane " 2 T)
(" is " (MODIFIER)
"in the merchantlane " 2 T)))

(PUTPROPS SUCCESSOR PRINFORMS ((3 " is " (MODIFIER)
"the successor (in time) of " 2 T)))

(PUTPROPS COURSE PRINFORMS ((3 " is " (MODIFIER)
"the course of " 2 T)
(3 " is " (MODIFIER)
"the course of ")))

(PUTPROPS ROUGHLY-THE-SAME-COURSE-AS PRINFORMS ((3 " is " (MODIFIER)
"roughly the same course as "
2 T)))

(PUTPROPS ROUGHLY-THE-SAME-SPEED-AS PRINFORMS ((3 " is " (MODIFIER)
"roughly the same speed as "
2 T)))

(PUTPROPS ID PRINFORMS ((2 " is " (MODIFIER)
3 T)))

(PUTPROPS ID-AMPLIFY PRINFORMS ((2 " is " (MODIFIER)
3 T)))

(PUTPROPS LOCATION PRINFORMS (("The location of " 2 " is " (MODIFIER)
3 T)))

(PUTPROPS TO-PORT PRINFORMS ((3 " is " (MODIFIER)
"the destination port of " 2 T)))

(PUTPROPS FROM-PORT PRINFORMS ((3 " is " (MODIFIER)
"the starting port of " 2 T)))

(PUTPROPS SAME-AS PRINFORMS ((2 " is " (MODIFIER)
"the same as " 3 T)))

(PUTPROPS PATROL PRINFORMS ((2 " is " (MODIFIER)
"a patrol" T)))

(PUTPROPS POSSIBLE-REPORT PRINFORMS (("One of the reports from " 3
" concerns "
2 T)))

(PUTPROPS CROSSPATHS PRINFORMS (("The path from " 2 " to " 3 T " does "

(MODIFIER)

"cross the path from "
T 4 " to " 5 T))

(PUTPROPS GRAZE PRINFORMS (("The path from " 2 " to " 3 T " does "
(MODIFIER)
"graze the path from" T 4
" to "
5 T)))

(PUTPROPS WENT-BEFORE PRINFORMS (("A ship moving from " 4 " to " 2 T
" between the times "
5 " and " 3 T
"could "
(MODIFIER)

from " "have avoided sighting by a patrol travelling
T 6 " to " 8
" between "
7 " and " 9
"by traversing the patrol viewing area before the flight"
T)))

(PUTPROPS WENT-AFTER PRINFORMS (("A ship moving from " 4 " to " 2 T
" between the times "
5 " and " 3 T
"could "
(MODIFIER)

from " "have avoided sighting by a patrol travelling
T 6 " to " 8
" between "
7 " and " 9 T
"by traversing the patrol viewing area after the flight"
T)))

(PUTPROPS BLOCKED-FROM PRINFORMS (("A passage from " 2 " to " 3 " is "
(MODIFIER)
"counterindicated"
T)))

(PUTPROPS DISSIMILAR PRINFORMS ((2 " is " (MODIFIER)
"dissimilar to " 3 T)))

(PUTPROPS SWR PRINFORMS (("A ship at " 2 " at time " 3 " could " T
(MODIFIER)
"reach " 4 " at time " 5 T
" by travelling at top speed (or less)"
T)))

(PUTPROPS SIMPLY-WITHIN-REACH PRINFORMS ((2 " is " (MODIFIER)
"within travel distance of "

3 T)))

```
(PUTPROPS WITHIN-REACH PRINFORMS ((2 " is " (MODIFIER)
                                   "reachable from " 3 T
                                   "even considering possible patrol overflights"
                                   T)))
```

```
(PUTPROPS NOT-FIRST PRINFORMS ((2 " is " (MODIFIER)
                                   "other than a first sighting of its platform"
                                   T)))
```

```
(PUTPROPS NOT-LAST PRINFORMS ((2 " is " (MODIFIER)
                                   "other than a last sighting of its platform"
                                   T)))
```

```
(PUTPROPS ALIAS PRINFORMS ((3 " is " (MODIFIER)
                               "really " 2 T)))
```

```
(PUTPROPS COURSEFROM PRINFORMS (("The course from " 2 " to " 3 " is "
                                (MODIFIER)
                                4 T)))
```

```
(PUTPROPS SPEEDFROM PRINFORMS (("To move from " 2 " to " 4 T "between "
                                3 " and " 5
                                " implies a speed of "
                                6 T)))
```

```
(RPAQQ STATES (<EXPLTREE> <PLATIS> <VALIS> <ATTIS> <TYPIS> <IDIS>
               <IDAMPIS> <WHATFORM> <WHOSE2FORM> <WHOSEFORM> <TELLABT>
               <WHEREFORM> <WHEREITEM> <WHAT2FORM> <TYPE2> <ID2> <IDAMP2>
               <OCCURNUM> <OTHER2>))
```

```
(PUTPROPS <EXPLTREE> QHPRODS ((Q "-uit" : (PROGN (TERPRI)
                                                    (PRIN1
                                                     "Leaving EXPLAIN")
                                                    (TERPRI)
                                                    (SETQ DONEFLG T)))
                               (SAVE "memory" : (PROGN (TERPRI)
                                                         (PRIN1
                                                          "On file: ")
                                                         (MEMSAVE
                                                          (READ))
                                                         (CLEARBUF)
                                                         (TERPRI)))
                               (NEW "rule" : (PROGN (TERPRI)
                                                       (DEFINEPD)
                                                       (APPLYRULE
                                                        (CAR PRODUCTIONS))
                                                       (CLEARBUF)
                                                       (TERPRI)))
                               (CHANGE "confidence in the rule"
                                       !RULENAME (= RN)
                                       :
                                       (CHANGECON RN))
                               (BREAK : (PROGN (BREAK1 NIL T Explain))
```

```

                                (TERPRI)
                                (CLEARBUF)))
[DISPLAY : (COND (DSPLAYFLG (DSPTOP))
                  (T (TERPRI)
                     (PRIN1
                      "Sorry, but the display is not enabled."))
              (TERPRI]
  (IS (<PLATIS> <VALIS> <ATTIS>))
  (WHY "is" !ASSERTION (= NODE)
    :
    (IMPLIESASRT NODE))
  (HOW "does rule" !RULE (= RUL)
    "apply to " !ASSERTION (= NODE)
    :
    (RULEXP RUL NODE))
  (WHAT (IS ARE)
    <WHAT2FORM>
    (= WHATANS)
    :
    (PRETTYANS WHATANS))
  (WHOSE <WHOSEFORM> (= WHOSEANS)
    :
    (PRETTYANS WHOSEANS))
  (TELL "me about" <TELLABT>)
  (HELP : (HLPEXPLN))
  (WHERE <WHEREFORM>)
  (WHO "is" (~ A)
    (<TYPE2> <ID2> <IDAMP2> <OTHER2>))
  (REPORT : (RECAPCONCS))))

(PUTPROPS <PLATIS> QHPRODS [(!PLATFORM (= PLAT)
                                     (~ A AN)
                                     (<TYPIS> <IDIS> <IDAMPIS>)
                                     (= WHAF)
                                     :
                                     (YESNO (JUGGLE WHAF PLAT)])

(PUTPROPS <VALIS> QHPRODS [(<WHOSE2FORM> !WHOSE2RES (= VAL58)
                             :
                             (YESNO (JUGGLE WHOSE2RES2 VAL58)])

(PUTPROPS <ATTIS> QHPRODS [(<WHATFORM> !WHATRES (= VAL57)
                             :
                             (YESNO (APPEND WHATRES2 (CONS VAL57)])

(PUTPROPS <TYPIS> QHPRODS ((!TYPE (= TYPEN)
                                :
                                (LIST (QUOTE TYPE)
                                     TYPEN))))

(PUTPROPS <IDIS> QHPRODS ((!ID (= IDN)
                             :
                             (LIST (QUOTE ID)
                                     IDN))))

(PUTPROPS <IDAMPIS> QHPRODS ((!ID-AMP (= IDA)
                                :

```

(LIST (QUOTE ID-AMPLIFY)
IDA)))

```
(PUTPROPS <WHATFORM> QHPRODS ((THE !RELATION (= RELNM)
                                "OF" !OBJECT (= OBJ)
                                :
                                (WHATFORMFN RELNM OBJ))))
```

```
(PUTPROPS <WHOSE2FORM> QHPRODS ((!VALUE (= VAL59)
(A AN THE a an the)
!RELATION
(= RELN4)
:
(WHOSE2FORMFN VAL59 RELN4))))
```

[illegible]

```
(PUTPROPS <TELLABT> QHPRODS [(!RELATION !OBJECT !VALUE)
  (= ITEMN)
  :
  (PROG (ISONE)
    (TERPRI)
    [MAPC ASSERTIONS (FUNCTION (LAMBDA
      (X)
      (COND
        ((MEMB ITEMN
          (GETUPLE X))
          (PRETTYASSR X)
          (SETQ ISONE T)

        (COND ((NOT ISONE)
          (PRIN1 "No information about that.")
          (TERPRI)
          (!RULENAME (= RUNM)
            :
            (PROGN (TERPRI)
              (FANCYPROD RUNM))
```

(TERPRI))

```

(PUTPROPS <WHEREITEM> QHPRODS [(!PLATFORM
    (= PLTNM)
    :
    (PROGN (TERPRI)
        (SETQ EXPLAINFLAG T)
        (PRIN1 (PLATPOS PLTNM CURTIME))
        (SETQ EXPLAINFLAG NIL)
        [COND (DSPLAYFLG (TERPRI)
            (PRIN1 "Also, see the display.")
            (TERPRI)
            (WAITER)
            (DSPCMD (CONCAT "PTR " PLTNM)
                (TERPRI)))
        ((!MLANE !STORM)
            (= LANM)
            :
            (PROGN (TERPRI)
                [PRIN1 (CDADAR (RETRIEVER (LIST (QUOTE LOCATION)
                    LANM
                    (QUOTE *LOC)
                    (COND (DSPLAYFLG (TERPRI)
                        (PRIN1 "Also, see the display.")
                        (TERPRI)
                        (WAITER)
                        (DSPCMD (CONCAT "PTR " LANM)
                            (TERPRI))
                (TERPRI))

```

```

(PUTPROPS <WHAT2FORM> QHPRODS (((<WHATFORM>)
    (!PLATFORM (= PA)
    :
    (WHAT2FORMFN PA))))

```

```

(PUTPROPS <TYPE2> QHPRODS [(!TYPE (= TYP)
    :
    (PRETTYANS
        (RETRIEVER (LIST (QUOTE TYPE)
            (QUOTE *WHO)
            TYP]))

```

```

(PUTPROPS <ID2> QHPRODS [(!ID (= IDB)
    :
    (PRETTYANS (RETRIEVER
        (LIST (QUOTE ID)
            (QUOTE *WHO)
            IDB))

```

```

(PUTPROPS <IDAMP2> QHPRODS [(!ID-AMP (= IDM)
    :
    (PRETTYANS
        (RETRIEVER (LIST (QUOTE
            ID-AMPLIFY)
            (QUOTE *WHO)
            IDM))

```

```

(PUTPROPS <OCCURNUM> QHPRODS ((!SMALLNUMB (= TIMES)

```

```

:
(OCCURPRINT TIMES NODE)))

```

```

(PUTPROPS <OTHER2> QHPRODS [(IRELATION
  (= RM)

```

```

:
(PRETTYANS (APPEND (RETRIEVER (LIST RM (QUOTE *WHO)))
  (RETRIEVER (LIST RM (QUOTE *WHO)
    (QUOTE *IGN]))

```

```

(RPAQQ NEWEXPENS (ASSRPRINT CHANGECON EXPLAIN GAMF HLPXPLN IMPLIESASRT
  JUGGLE MAKEPRINT MEMSAVE MODIFIER NEWVALOBJ
  NICEANSWER OCCURPRINT PRETTYANS PRETTYASSR
  PRINTRULEASSR RECAPCONCS RESOUT
  RESULTPRINTER RULEXP WHAT2FORMFN WHATFORMFN
  WHOSE2FORMFN YESNO DSPEXP))

```

```

(DEFINEQ

```

[148]

```

(ASSRPRINT
  [LAMBDA (PRINSPEC)

```

```

(* edited:
  "24-Aug-79 12:46")

```

```

(* This is the workhorse of the assertion
prettyprinter. It receives a PRINFORM as an
argument, and prints in accordance with what is
found there. Strings are printed as found.
Numbers refer to "slots" in the GETUPLE of the
assertion in question. Lists
(assumed to be functions) are evaluated and must do
their own printing. T causes a TERPRI.)

```

```

(COND
  ((STRINGP PRINSPEC)
    (PRIN1 PRINSPEC))
  [(NUMBERP PRINSPEC)
    (PRIN1 (CAR (NTH BODY PRINSPEC)
      ((AND (LISTP PRINSPEC)
        (OR (NULL LSTFLG)
          OVERCONF))
        (EVAL PRINSPEC))
      ((EQ PRINSPEC T)
        (PRIN1 ".")
        (TERPRI))

```

[149]

```

(CHANGECON
  [LAMBDA (RLNME1)

```

```

(* edited:
  " 7-Aug-79 08:19")

```

```

(* Allows the user to change the confidence in the
rule that is its argument. This change, while
permanent for that invocation of STAMMER
(until CHANGECON is called again), does not affect

```

future invocations unless the rules are saved
(by doing MAKEFILE (RULES)). The changed confidence
IS reflected in all inferences done before the
CHANGECON, due to the dynamic calculation of
confidence.)

```
(TERPRI)
(PRIN1 " Present confidence is: ")
(PRIN1 (GETPROP RLNMEL (QUOTE CONF)))
(TERPRI)
(PRIN1 " Confidence should be: ")
(PUTPROP RLNMEL (QUOTE CONF)
  (READ))
(CLEARBUF)
(TERPRI)
```

[150]

```
(EXPLAIN
  [LAMBDA NIL
```

```
(* edited:
  "24-Aug-79 17:55")
```

(* The top level of the explanation system.
Most of this function is initialization of variables
used by the explanation productions.
The most notable feature is the use of ERSETQ to
allow the user to escape back to the top level of
explanation via control-E if and when he/she gets
totally lost.)

```
(PROG (DONEFLG [PLATFORM (CONS (QUOTE CONSOLE)
  (RETRIEVES (QUOTE PLATFORM)
    (QUOTE *)]
  (MLANE (RETRIEVES (QUOTE MERCHANTLANE)
    (QUOTE *)))
  (STORM (RETRIEVES (QUOTE STORM)
    (QUOTE *)))
  (ID (QUOTE (FRIEND HOSTILE UNKNOWN)))
  (ID-AMP (QUOTE (NON-MIL MIL-BATTLE MIL-AUXIL UNKNOWN)
    ))
  (TYPE (QUOTE (CARRIER CRUISER DESTROYER FRIGATE
    AMPHIB-ASSAULT AMPHIB-DOCK
    PATROL-BOAT MINELAYER
    MINESWEEPER LANDING SUB OILER
    AMMUNITION STORES
    DESTROYER-TENDER SUB-TENDER
    BUOY-TENDER
    PATROL-CRAFT-TENDER REPAIR
    RESEARCH INTELLIGENCE TUG
    MERCHANT FISHING PASSENGER
    PLEASURE MISCELLANEOUS BOMBER
    FIGHTER RECONNISANCE)))
  VALUE OBJECT)
  (SETQ RULENAME PRODUCTIONS)
  (SETQ RELATION RELATIONS)
```

```

(MAPC ASSERTIONS (FUNCTION NEWVALOBJ))
LOOP(OR (ERSETQ (QHTAKE "Question? " <EXPLTREE>))
      (GO LOOP))
(COND
  (DONEFLG (RETURN))
  (T (GO LOOP)))

```

[151]

```

(GAMF
  [LAMBDA (WLK OVERRIDE)
    (PROG (CONFI ACON)

```

```

(* edited:
  " 8-Aug-79 19:49")

```

```

(* GAMF generates an appropriate modifier for an
assertion based on the confidence of the assertion.)

```

```

(SETQ CONFI (OR OVERRIDE (GETCON WLK)))
(SETQ ACON (ABS CONFI))
(COND
  ((EQP ACON 1.0))
  ((FGREATERP ACON .98)
   (PRIN1 "definitely "))
  ((FGREATERP ACON .9)
   (PRIN1 "almost certainly "))
  ((FGREATERP ACON .7)
   (PRIN1 "very probably "))
  ((FGREATERP ACON .45)
   (PRIN1 "probably "))
  ((EQP ACON 0.0)
   (PRIN1 "not known to be ")
   (RETURN))
  (T (PRIN1 "somewhat ")
   (COND
     ((FLESSP CONFI 0.0)
      (PRIN1 "un")))
     (PRIN1 "likely to be ")
     (RETURN)))
  (COND
    ((FLESSP CONFI 0.0)
     (PRIN1 "not "))

```

[152]

```

(HLPEXPLN
  [LAMBDA NIL

```

```

    (PRIN1 "Sorry, no help yet.")
    (TERPRI))

```

```

(* edited:
  "24-Jul-79 18:56")

```

[153]

```

(IMPLIESASRT
  [LAMBDA (NODE)
    (PROG (X)

```

```

(* edited:
  "17-Aug-79 14:14")

```

```

(TERPRI)
(COND
  ((GETPROP NODE (QUOTE TDB))
    (PRIN1
      "That assertion is part of the technical data base")
    (TERPRI)
    (RETURN))
  (T))
(SETQ X (GETPROP NODE (QUOTE DERIVE*)))
(COND
  ((AND (NULL X)
    (EQP (GETCON NODE)
      0))
    (PRIN1 "Assertion based on the absence of information")
    (TERPRI))
  ((GETPROP (CAR (GETUPLE NODE))
    (QUOTE ORACLE))
    (PRIN1 "That assertion was computed by the oracle ")
    (PRIN1 (CAR (GETUPLE NODE)))
    (TERPRI))
  (X (PRIN1 "STAMMER applied the rule(s)")
    (TERPRI)
    [MAPC X
      (FUNCTION (LAMBDA (Y)
        (PROGN [COND
          ((MEMBER (CAR Y)
            RULE))
          (T (SETQ RULE
            (APPEND (LIST (CAR Y))
              RULE]
            (PRIN1 (CAR Y))
            (SPACES 1]
            (TERPRI))
      (T (PRIN1 "The information came directly from a message.")
        (TERPRI))

```

[154]

```

(JUGGLE
  [LAMBDA (PAIR INSERTITEM)

```

```

(* edited:
  " 7-Aug-79 08:24")

```

```

(* JUGGLE return a three element list constructed by
  placing INSERTITEM between the elements of PAIR.
  It's non-destructive, and costs due to
  (possibly unnecessary) copying.)

```

```

(LIST (CAR PAIR)
  INSERTITEM
  (CADR PAIR))

```

[155]

```

(MAKEPRINT
  [LAMBDA (RELN)

```

```

(* edited:
  " 7-Aug-79 08:28")

```

(* MAKEPRINT is provided as an assistance in creating PRINFORMS. Given a relation name, MAKEPRINT prompts for new PRINFORMS, after printing the existing forms, if any. To terminate addition, type STOP.)

```
(PROG (PFORM NEWFORM)
  MP1 (PRIN1 "For the relation ")
      (PRIN1 RELN)
      (TERPRI)
      (PRIN1 "use the prinforms ")
      (SETQ PFORM (GETPROP RELN (QUOTE PRINFORMS)))
      [COND
        (PFORM (MAPC PFORM (FUNCTION (LAMBDA (X)
                                          (TERPRI)
                                          (SPACES 3)
                                          (PRIN1 X)
                                        )
                                )
          )
        )
      ]
  MPLP (TERPRI)
      (SPACES 3)
      (SETQ NEWFORM (READ))
      (COND
        ((EQ NEWFORM (QUOTE STOP))
          (GO EXLOOP))
        )
      (SETQ PFORM (APPEND PFORM (CONS NEWFORM)))
      (GO MPLP)
  EXLOOP
    (PUTPROP RELN (QUOTE PRINFORMS)
              PFORM)
    (TERPRI))
```

[156]

```
(MEMSAVE
  [LAMBDA (FEE)
```

```
(* edited:
  " 7-Aug-79 08:30")
```

(* MEMSAVE saves the contents of memory (exclusive of stream suspensions) on a user specified file. This is made simple since memory can be set up completely through standard fileCOMS, which are assigned to the user file name.)

```
(SET (FILECOMS FEE)
      MEMORYCOMS)
(MAKEFILE FEE)
(TERPRI)
(PRIN1 "Memory saved.")
(TERPRI)
(CLEARBUF)
```

[157]

```
(MODIFIER
  [LAMBDA NIL
```

```
(* edited:
  "22-Aug-79 20:04")
```

```
(* MODIFIER provides a way for GAMF to be used in
  PRINFORMs while allowing the user to remain
  blissfully unaware of how to refer to the node he's
  describing. Since NODE is used freely here, MODIFIER
  should ONLY be used in PRINFORMs, never as a
  standard function call.)
```

```
(PROG (CON)
  (GAMF NODE OVERCONF)
  (COND
    ([AND (NOT OVERCONF)
          (NOT (MEMBER (SETQ CON (GETCON NODE))
                      (QUOTE (1.0 0.0 -1.0))
          (PRIN1 "(")
          (PRIN1 (TWO-PLACE CON))
          (PRIN1 ") "])
```

[158]

```
(NEWVALOBJ
  [LAMBDA (ARRT)
```

```
(* edited:
  "24-Aug-79 12:26")
```

```
(* This function sets up lists of object and value
  slot fillers that are presently used in memory.
  These lists are then used by the explanation system
  productions. In a standard TWOARG assertion, the
  format of the assertion is (REL OBJ VAL).)
```

```
(PROG (VL OJ TUPLE)
  (COND
    ((LESSP (LENGTH (SETQ TUPLE (GETUPLE ARRT)))
            3)
    (RETURN)))
  (SETQ VL (CADDR TUPLE))
  (SETQ OJ (CADR TUPLE))
  [COND
    ((LISTP VL))
    ((NUMBERP VL))
    ((MEMB VL VALUE))
    (T (SETQ VALUE (CONS VL VALUE))
  (COND
    ((LISTP OJ))
    ((NUMBERP OJ))
    ((MEMB OJ OBJECT))
    (T (SETQ OBJECT (CONS OJ OBJECT))
```

[159]

```
(NICEANSWER
  [LAMBDA (ANS1)
```

```
(* edited:
  " 7-Aug-79 08:38")
```

```
(* Ocassionally, you don't want to print a whole
  assertion, but rather just a value or object, but
  with a confidence indicator attached.
  NICEANSWER does this. It assumes that its argument
  is a single element of a RETRIEVER answer.)
```

```
(GAMF (CAR ANS1))
(PRIN1 (CDADR ANS1))
(TERPRI)
```

[160]

```
(OCCURPRINT
  [LAMBDA (TIMES NODE)
```

```
(* edited:
  "24-Aug-79 17:54")
```

```
(PROG (X Z)
  (SETQ X (GETPROP NODE (QUOTE DERIVE*)))
  [for Y in X UNTIL (ZEROP TIMES)
    do (COND
      ((EQUAL (CAR Y)
               RULE)
        (SETQ TIMES (SUB1 TIMES))
        (SETQ Z Y)
        (PRINTRULEASSR Z)
        (QHTAKE "Another occurrence?" (!carriagereturn <OCCURNUM>))
```

[161]

```
(PRETTYANS
  [LAMBDA (ANSLST)
```

```
(* edited:
  " 7-Aug-79 08:39")
```

```
(* PRETTYANS gets whatever RETRIEVER returns and
  uses NICEANSWER to print the results, if any.
  If there are no results, PRETTYANS admits
  ignorance.)
```

```
(TERPRI)
(COND
  ((NULL ANSLST)
   (PRIN1 "I don't know.")
   (TERPRI))
  (T (MAPC ANSLST (FUNCTION NICEANSWER))
```

[162]

```
(PRETTYASSR
  [LAMBDA (NODE FORMAT OVERCONF)
```

```
(* edited:
  "24-Aug-79 12:45")
```

(* PRETTYASSR is the assertion prettyprinter.
 Every relation is assumed to have a list of
 PRINFORMS on its property list that will be used to
 guide the printing of assertions with that relation.
 PRETTYASSR is called on an assertion with a selector
 as to which PRINFORM to use.
 The default PRINFORM is the first.
 If there are no PRINFORMS stored, defaults are used,
 but their beauty is not guaranteed.)

```
(PROG (BODY FORMLST USEFORM LSTFLG)
[COND
  ((LISTP NODE)
    (SETQ LSTFLG T)
    (SETQ BODY NODE))
  (T (PRIN1 NODE)
    (PRIN1 ": ")
    [COND
      ((MEMB NODE ASSERTION))
      (T (SETQ ASSERTION (CONS NODE ASSERTION)
        (SETQ BODY (EVAL NODE))
      (COND
        ((NULL FORMAT)
          (SETQ FORMAT 1)))
      (SETQ FORMLST (GETPROP (CAR BODY)
        (QUOTE PRINFORMS)))
      (COND
        ((GREATERP FORMAT (LENGTH FORMLST))
          (SETQ FORMAT 1)))
      [COND
        [(NULL FORMLST)
          (SELECTQ (LENGTH BODY)
            [2 (SETQ USEFORM (QUOTE (2 " is " (MODIFIER)
              " a " 1 T]
            [3 (SETQ USEFORM (QUOTE (3 " is " (MODIFIER)
              " a " 1 " of " 2 T]
            (SETQ USEFORM (FOR I FROM 1 TO (LENGTH BODY)
              COLLECT I]
          (T (SETQ USEFORM (CAR (NTH FORMLST FORMAT]
        (MAPC USEFORM (FUNCTION ASSRPRINT])
```

[163]

```
(PRINTRULEASSR
  [LAMBDA (RULEASSRTS)
```

```
(* edited:
  "17-Aug-79 13:24")
```

```
(TERPRI)
(COND
  [RULEASSRTS (PRIN1 "The rule was applied with the assertions")
    (TERPRI)
    [for Y in (CDR RULEASSRTS)
      do (TERPRI)
        (COND
          [(ATOM Y)
            (COND
```

```

((FLESSP 0 (GETCON Y))
 (PRETTYASSR Y))
(T (PRETTYASSR Y NIL .4)
  (TAB 7)
  (PRIN1 "(condition is no longer true)")
  (TERPRI))
(T (COND
   [(EQ (CAR Y)
        (QUOTE NOT))
    (COND
     ((FLESSP (GETCON (CADR Y))
              0)
      (PRETTYASSR (CADR Y)))
     (T (PRETTYASSR (CADR Y)
                    NIL -.4)
        (TAB 7)
        (PRIN1 "(no longer valid)")
        (TERPRI))
      ((EQ (CAR Y)
           (QUOTE UNLESS))
       (COND
        ((FLESSP 0 (GETCON (CADR Y)))
         (PRETTASSR (CADR Y)
                    NIL 0.0)
         (TAB 7)
         (PRIN1 "(no longer valid)")
         (TERPRI))
        (T (PRETTYASSR (CADR Y)
                       (COND
                        ((AND DSPLAYFLG (DSPEXP RULEASSRTS))
                         (TERPRI)
                         (PRIN1 "Also, see the display.")
                         (TERPRI)
                         (WAITER)
                         (DSPCMD (DSPEXP RULEASSRTS)))
                        (T (TERPRI))
                        (T (PRIN1 "The rule was not applied to derive that assertion")
                         (TERPRI))

```

[164]

```

(RECAPCONCS
 [LAMBDA NIL

```

```

(* edited:
 " 3-Aug-79 14:07"

```

```

(TERPRI)
(TERPRI)
(MAPC ASSERTION (FUNCTION PRETTYASSR))

```

[165]

```

(RESOUT
 [LAMBDA NIL

```

```

(* edited:
 " 7-Aug-79 07:49"

```

(* RESOUT causes the results of rule firings to be printed at the user's terminal, using the function RESULTPRINTER. First, it removes duplications of

conclusions (so a single conclusion is printed only once).)

```
(SETQ RESULTLIST (INTERSECTION RESULTLIST RESULTLIST))
(MAPC RESULTLIST (FUNCTION RESULTPRINTER))
(SETQ RESULTLIST NIL)
```

[166]

```
(RESULTPRINTER
  [LAMBDA (RES1)
```

```
(* edited:
  "17-Aug-79 17:22")
```

(* Results can either be a report, in which case a report flag is printed and then the report itself is printed (literally), or assertions, which are handled by the assertion prettyprinter.)

```
(COND
  ((LISTP RES1)
    (PRIN1 "Report: ")
    (MAPC (CDR RES1)
      (FUNCTION PRIN1))
    (TERPRI))
  ((NOT (MEMB (CAR (GETUPLE RES1))
    DULLREL))
    (PRETTYASSR RES1))
```

[167]

```
(RULEXP
  [LAMBDA (RULE NODE)
```

```
(* edited:
  " 9-Aug-79 11:49")
```

```
(PROG (X Z COUNT)
  (COND
    ((GETPROP RULE (QUOTE ORACLE))
      (PRIN1 "That assertion was computed by an oracle")
      (TERPRI)
      (RETURN))
    ((GETPROP NODE (QUOTE TDB))
      (PRIN1 "That assertion came from the
technical data base")
      (TERPRI)
      (RETURN))
    (T))
  (SETQ X (GETPROP NODE (QUOTE DERIVE*)))
  (COND
    ((NULL X)
      (PRIN1 "That assertion came from a message.")
      (TERPRI)
      (RETURN)))
  (SETQ COUNT 2)
  [for Y in X until (ZEROP COUNT)
    do (COND
      ((EQUAL (CAR Y)
```

```

                RULE)
            (SETQ COUNT (SUB1 COUNT))
            (SETQ Z Y)
(COND
  ((EQUAL COUNT 1)
   (PRINTRULEASSR Z))
  ((ZEROP COUNT)
   (TERPRI)
   (QHTAKE "Which occurrence?" <OCCURNUM>))
  (T (PRIN1
      "The rule was not applied to derive that assertion")
      (TERPRI)))

```

[168]

```

(WHAT2FORMFN
 [LAMBDA (PL)

```

```

(* edited:
 " 7-Aug-79 08:48")

```

```

(* This function collects answers to the question
"what is <some platform>" by looking in the memory
for the things a platform can be.)

```

```

(APPEND (RETRIEVER (LIST (QUOTE ID)
                          PL
                          (QUOTE *WHA)))
  (RETRIEVER (LIST (QUOTE ID-AMPLIFY)
                  PL
                  (QUOTE *WHA)))
  (RETRIEVER (LIST (QUOTE TYPE)
                  PL
                  (QUOTE *WHA)))
  (RETRIEVER (LIST (QUOTE CLASS)
                  PL
                  (QUOTE *WHA)))

```

[169]

```

(WHATFORMFN
 [LAMBDA (REL OBJ)

```

```

(* edited:
 " 7-Aug-79 08:50")
(* This generates
answers to the other
form of WHAT questions,
e.g.
"what is the rel of obj?"

```

```

)
(PROG (ANS)
  [SETQ ANS (RETRIEVER (LIST REL OBJ (QUOTE *VAL)
                             (SETQ WHATRES2 (LIST REL OBJ)
                             (SETQ WHATRES (MAPCAR ANS (FUNCTION CDADR)))
                             (RETURN ANS))

```

[170]

```
(WHOSE2FORMFN
  [LAMBDA (VAL REL)
```

```
(* edited:
  " 7-Aug-79 08:51")
(* For getting answers
  to the question
  "whose rel is val?")
```

```
(PROG (ANS)
  (SETQ ANS (RETRIEVER (LIST REL (QUOTE *WHO)
                                VAL)))
  (SETQ WHOSE2RES2 (LIST REL VAL))
  (SETQ WHOSE2RES (MAPCAR ANS (FUNCTION CDADR)))
  (RETURN ANS))
```

[171]

```
(YESNO
  [LAMBDA (ASSRSPEC)
```

```
(* edited:
  " 7-Aug-79 08:52")
```

```
(* If you don't want to print an entire assertion or
  even a part, but just want to answer yes, no, or
  some confidence modifier (like in response to "is"
  questions).)
```

```
(PROG ((NDE (CAR (GETSTRIP ASSRSPEC)))
  NDECON)
  (TERPRI)
  (SETQ NDECON (GETCON NDE))
  (COND
    ((EQP NDECON 1.0)
     (PRIN1 "Yes"))
    ((EQP NDECON -1.0)
     (PRIN1 "No"))
    ((EQP NDECON 0.0)
     (PRIN1 "I don't know"))
    (T (PRIN1 "It's ")
       (GAMF NDE)))
  (TERPRI))
```

.172]

```
(DSPEXP
  [LAMBDA (BOX)
```

```
(* edited:
  " 8-Aug-79 15:47")
```

```
(PROG ((DSPOBJECTS (QUOTE (PLATFORM CONTACT STORM MERCHANTLANE
  OWNSHIP)))
  (DSPLST (CONS (QUOTE PTR)))
  (BLANK " ")
  (COMMA ",")
  (COUNT 0))
  [for X in (CDR BOX)
    do [COND
      ((LISTP X)
       (SETQ X (CADR X)
```

```
(* To deal with UNLESS's
```

```

etc.)
(SETQ X (GETUPLE X))
(COND
  ((OR (MEMB (CAR X)
             DSPOBJECTS)
        (EQ (CAR X)
             (QUOTE RANGE))))
    (SETQ DSPLST (CONS (COND
                        ((ZEROP COUNT)
                         BLANK)
                        (T COMMA))
                        DSPLST))
      (SETQ DSPLST (CONS (COND
                          ((EQ (CAR X)
                               (QUOTE RANGE))
                           OWNSHIP)
                          (T (CADR X)))
                          DSPLST))
        (SETQ COUNT (ADD1 COUNT))
      (COND
        ((IGREATERP COUNT 1)
         (RETURN (APPLY (FUNCTION CONCAT)
                        (DREVERSE DSPLST)))
        )
      (LOAD (QUOTE QH.COM))
      (DECLARE: DONTCOPY
        (FILEMAP (NIL (14818 33280 (ASSRPRINT 14830 . 15593) (CHANGECON 15597 .
16372) (EXPLAIN 16376 . 17885) (GAMF 17889 . 18799) (HLPXPLN 18803 .
18949) (IMPLIESASRT 18953 . 20137) (JUGGLE 20141 . 20493) (MAKEPRINT
20497 . 21474) (MEMSAVE 21478 . 21971) (MODIFIER 21975 . 22724) (
NEWVALOBJ 22728 . 23634) (NICEANSWER 23638 . 24068) (OCCURPRINT 24072 .
24481) (PRETTYANS 24485 . 24893) (PRETTYASSR 24897 . 26379) (
PRINTRULEASSR 26383 . 27895) (RECAPCONCS 27899 . 28069) (RESOUT 28073 .
28551) (RESULTPRINTER 28555 . 29089) (RULEXP 29093 . 30225) (WHAT2FORMFN
30229 . 30790) (WHATFORMFN 30794 . 31269) (WHOSE2FORMFN 31273 . 31714)
(YESNO 31718 . 32358) (DSPEXP 32362 . 33277)))))
      STOP

```

(FILECREATED " 8-Aug-79 09:11:09" <DKIBLER>ORACLE.LSP.40 25437

changes to: SPEEDM

previous date: " 7-Aug-79 17:55:13" <DKIBLER>ORACLE.LSP.39)

(PRETTYCOMPRINT ORACLECOMS)

(RPAQQ ORACLECOMS [(VARS * ORACLEVARS)
 (IFPROP (ORACLE ORTYPE)
 * ORACLES)
 (FNS * ORACLEFNS)
 (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
 COMPILERVERS (ADDVARS (NLAMA WITHINR)
 (NLAML)
 (LAMA))

(RPAQQ ORACLEVARS (ORACLES MAXSHIPSPEED))

(RPAQQ ORACLES (SAME-AS ROUGHLY-THE-SAME-SPEED-AS
 ROUGHLY-THE-SAME-COURSE-AS IN-LANE INSIDE
 GREATER-THAN LESS-THAN CROSSPATHS GRAZE SWR
 WENT-AFTER WENT-BEFORE SUCCESSOR PREDECESSOR
 RANGE BEARING COURSE SPEED COURSEFROM SPEEDFROM)
)

(RPAQQ MAXSHIPSPEED 35)

(RPAQQ ORACLES (SAME-AS ROUGHLY-THE-SAME-SPEED-AS
 ROUGHLY-THE-SAME-COURSE-AS IN-LANE INSIDE
 GREATER-THAN LESS-THAN CROSSPATHS GRAZE SWR
 WENT-AFTER WENT-BEFORE SUCCESSOR PREDECESSOR
 RANGE BEARING COURSE SPEED COURSEFROM SPEEDFROM)
)

(PUTPROPS SAME-AS ORACLE T)

(PUTPROPS ROUGHLY-THE-SAME-SPEED-AS ORACLE T)

(PUTPROPS ROUGHLY-THE-SAME-COURSE-AS ORACLE T)

(PUTPROPS IN-LANE ORACLE T)

(PUTPROPS INSIDE ORACLE T)

(PUTPROPS GREATER-THAN ORACLE T)

(PUTPROPS LESS-THAN ORACLE T)

(PUTPROPS CROSSPATHS ORACLE T)

(PUTPROPS GRAZE ORACLE T)

(PUTPROPS SWR ORACLE T)

(PUTPROPS WENT-AFTER ORACLE T)

(PUTPROPS WENT-BEFORE ORACLE T)

(PUTPROPS SUCCESSOR ORACLE T)

(PUTPROPS PREDECESSOR ORACLE T)

(PUTPROPS RANGE ORACLE T)

(PUTPROPS BEARING ORACLE T)

(PUTPROPS COURSE ORACLE T)

(PUTPROPS SPEED ORACLE T)

(PUTPROPS SUCCESSOR ORTYPE LASTARG)

(PUTPROPS PREDECESSOR ORTYPE LASTARG)

(PUTPROPS RANGE ORTYPE LASTARG)

(PUTPROPS BEARING ORTYPE LASTARG)

(PUTPROPS COURSE ORTYPE LASTARG)

(PUTPROPS SPEED ORTYPE LASTARG)

(PUTPROPS COURSEFROM ORTYPE LASTARG)

(PUTPROPS SPEEDFROM ORTYPE LASTARG)

(RPAQQ ORACLEFNS (SAME-AS ROUGHLY-THE-SAME-SPEED-AS
ROUGHLY-THE-SAME-COURSE-AS IN-LANE INSIDE
GETATTB GREATER-THAN LESS-THAN BEARING SPEED
INTERIOR DISTANCE DISTOLINE INLANE LINPOLY
CROSSBOUNDARY SOMELINESEG TRACKINPOLY
CROSSLINES OPSIDES ROTSENSE SUBTEND LANERANGE
WITHINR CROSSPATHS LOCATION POSS-REPORT
DISSIMILPLAT WENT-BEFORE WENT-AFTER LOC-TIME
SWR SPEEDM GRAZE SUCCESSOR PREDECESSOR
DIRECTION RANGE COURSE SPEEDAUX COURSEFROM
SPEEDFROM))

(DEFINEQ

[173]

(SAME-AS
[LAMBDA (W U)

(EQ W U))

(* edited:
"25-Jul-79 19:16")

[174]

(ROUGHLY-THE-SAME-SPEED-AS
[LAMBDA (Q1 Q2)

(* edited:
" 6-Aug-79 20:31")

(* Speeds are considered to be roughly the same if they are within 5 per cent of each other.)

```
(AND (GREATERP (PLUS Q2 (TIMES Q2 .05))
              Q1)
      (GREATERP Q1 (DIFFERENCE Q2 (TIMES Q2 .05)))
```

[175]

```
(ROUGHLY-THE-SAME-COURSE-AS
 [LAMBDA (Q1 Q2)
```

(* edited:
" 6-Aug-79 19:20")

(* Two courses are considered to be roughly the same if they differ by at most 4.5 percent.)

```
(AND (GREATERP (PLUS Q2 4.5)
              Q1)
      (GREATERP Q1 (DIFFERENCE Q2 4.5)))
```

[176]

```
(IN-LANE
 [LAMBDA (MLANE POS)
```

(* edited:
" 6-Aug-79 19:24")

(* If the centroid of the position is within 5 nautical miles of the given lane, this function returns true.)

```
(PROG ((Y (CENTROID POS))
       (X (LAST MLANE)))
      (RETURN (FGREATERP 5.0 (LANERANGE (CAAR MLANE)
                                         (CADAR MLANE)
                                         (CAAR X)
                                         (CADAR X)
                                         (CAR Y)
                                         (CADR Y))
```

[177]

```
(INSIDE
 [LAMBDA (POS STORM)
```

(* edited:
" 6-Aug-79 19:28")

(* This function returns true if the centroid of the position is interior to the polygon.)

```
(APPLY (FUNCTION INTERIOR)
      (APPEND (CENTROID POS)
              (CONS STORM)))
```

[178]

```
(GETATTB
  [LAMBDA (REL NODE)
    (* edited:
      "26-Jul-79 18:22")
    (PROG [(SPEC (QUOTE (SIGHTING * SIGHTING3)
      (RPLACA SPEC REL)
      (RPLACA (CDR SPEC)
        (QUOTE *))
      (RPLACA (CDDR SPEC)
        NODE)
      (RETURN (CADR (GETUPLE (CAR (STRIPSTREAM (GETSH SPEC))
```

[179]

```
(GREATER-THAN
  [LAMBDA (Q1 Q2)
    (* edited:
      "25-Jul-79 13:55")
    (GREATERP Q1 Q2))
```

[180]

```
(LESS-THAN
  [LAMBDA (Q1 Q2)
    (* edited:
      "25-Jul-79 13:56")
    (GREATERP Q2 Q1))
```

[181]

```
(BEARING
  [LAMBDA (SITE)
    (* edited:
      " 6-Aug-79 20:13")
```

(* Bearing accepts a sighting node and returns the bearing from the ownship to the sighted platform.)

```
(PROG (POS1 POS2 TIME)
  (SETQ TIME (GETATT (QUOTE TOS)
    SITE))
  (SETQ POS1 (OWNPOS TIME))
  (SETQ POS2 (CENTROID (GETATT (QUOTE POSITION)
    SITE)))
  (RETURN (DIRECTION (CAR POS1)
    (CADR POS1)
    (CAR POS2)
    (CADR POS2)))
```

[182]

```
(SPEED
  [LAMBDA (SITE)
    (* edited:
      " 7-Aug-79 12:49")
```

(* Speed accepts a sighting node and computes an estimated speed using the closer of the predecessor or successor.)

```

(PROG (PRED SUC TPRED TSUC PPRED PSUC POS TIME)
  (SETQ TIME (GETATT (QUOTE TOS)
                     SITE))
  (SETQ POS (CENTROID (GETATT (QUOTE POSITION)
                              SITE)))
  (SETQ PRED (PREDECESSOR SITE))
  (SETQ SUC (SUCCESSOR SITE))
  [COND
    (SUC (SETQ TSUC (GETATT (QUOTE TOS)
                           SUC))
          (SETQ PSUC (CENTROID (GETATT (QUOTE POSITION)
                                       SUC)
                              SUC)
          [COND
            (PRED (SETQ TPRED (GETATT (QUOTE TOS)
                                       PRED))
                  (SETQ PPRED (CENTROID (GETATT (QUOTE POSITION)
                                                PRED)
                                       PRED)
            (COND
              ((AND (NULL PRED)
                    (NULL SUC)
                     (RETURN)))
              ((NULL PRED)
               (RETURN (SPEEDM TIME TSUC (DISTANCE (CAR POS)
                                                    (CADR POS)
                                                    (CAR PSUC)
                                                    (CADR PSUC)
                                                    )
                    ((NULL SUC)
                     (RETURN (SPEEDM TPRED TIME (DISTANCE (CAR PPRED)
                                                            (CADR PPRED)
                                                            (CAR POS)
                                                            (CADR POS)
                                                            )
                           ((LESSP (FDIFFERENCE TIME TPRED)
                                    (FDIFFERENCE TSUC TIME))
                            (RETURN (SPEEDM TPRED TIME (DISTANCE (CAR PPRED)
                                                                    (CADR PPRED)
                                                                    (CAR POS)
                                                                    (CADR POS)
                                                                    )
                                (T (RETURN (SPEEDM TIME TSUC (DISTANCE (CAR POS)
                                                                        (CADR POS)
                                                                        (CAR PSUC)
                                                                        (CADR PSUC)

```

[183]

```

(INTERIOR
  [LAMBDA (OLAT OLON POLYGON)

```

```

(* edited:
  "30-Jul-79 10:54")

```

```

(* This function determines whether the point
  (OLAT OLON) is inside a polygon.
  The value of POLYGON must be a list of the vertices
  in either clockwise or counter-clockwise
  (starting anywhere) order. Each vertex is
  represented by a two element list containing the
  latitude and longitude.)

```

```

(PROG ((SUM 0.0)
      (POS1 (POLYGON:-1)))
      (SETN SUM 0.0)
      (* Must reinitialize SUM
      because of SETNs)
      (for POS in POLYGON
        do (PROG ((LAT (POS:1))
                  (LON (POS:2))
                  (LAT1 (POS1:1))
                  (LON1 (POS1:2))
                  (INC 0.0))
              (SETN INC ((DIRECTION OLAT OLON LAT LON)
                        -(DIRECTION OLAT OLON LAT1 LON1)))
              (if INC LT -180
                  then (SETN INC (INC+360))
                  elseif INC GT 180
                  then (SETN INC (INC-360)))
              (SETN SUM (SUM+INC))
              (POS1_POS)))
      (RETURN ((ABS SUM)
              GT 180))

```

[184]

```

(DISTANCE
 [LAMBDA (LAT1 LON1 LAT2 LON2)
 (FTIMES 60 (SUBTEND LAT1 LON1 LAT2 LON2))

```

[185]

```

(DISTOLINE
 [LAMBDA (X Y X1 Y1 X2 Y2)
 (* edited:
 "19-Jul-79 17:22")

```

(* Computes the distance from a given point to a
line segment between two given points)

```

(PROG ((A 1.369063E34)
      (B 1.369063E34)
      (C 1.369063E34)
      (COS1 0.0)
      (COS2 0.0))
      (SETN A (DISTANCE X Y X1 Y1))
      (SETN B (DISTANCE X Y X2 Y2))
      (SETN C (DISTANCE X1 Y1 X2 Y2))
      (SETN COS1 (FDIFFERENCE (FPLUS (FTIMES A A)
                                     (FTIMES C C))
                             (FTIMES B B))/(2*A*C))
      (SETN COS2 (FDIFFERENCE (FPLUS (FTIMES B B)
                                     (FTIMES C C))
                             (FTIMES A A))/(2*B*C))
      (RETURN (COND
              ((OR (MINUSP COS1)
                  (MINUSP COS2))
               (MIN A B))

```

```
(T (FTIMES A (SIN (ARCCOS (MIN 1 COS1))
```

[186]

```
(INLANE
```

```
  [LAMBDA (X Y LANE)
```

```
    (* NOBIND
```

```
    "15-Dec-78 14:06")
```

```
  (PROG ((X1 68.39)
```

```
        (Y1 -16.57)
```

```
        (X2 68.39)
```

```
        (Y2 -16.57))
```

```
  (SETN X1 LANE:1:1)
```

```
  (SETN Y1 LANE:1:2)
```

```
  (if [SOME LANE::1
```

```
      (FUNCTION (LAMBDA (LANEPOINT)
```

```
        (SETN X2 LANEPOINT:1)
```

```
        (SETN Y2 LANEPOINT:2)
```

```
        (PROG1 (LESSP (DISTOLINE X Y X1 Y1 X2 Y2)
                   MERCHANTLANEWIDTH)
```

```
          (SETN X1 X2)
```

```
          (SETN Y1 Y2)
```

```
      then (RETURN T))
```

[187]

```
(LINPOLY
```

```
  [LAMBDA (PT1 PT2 POLY)
```

```
    (* Checks if any part of
    line segment is in
    polygon)
```

```
  (OR (CROSSBOUNDARY PT1 PT2 POLY)
```

```
      (INTERIOR PT1:1 PT1:2 POLY))
```

[188]

```
(CROSSBOUNDARY
```

```
  [LAMBDA (PT1 PT2 POLY)
```

```
    (* Determines whether
    line from PT1 to PT2
    crosses the boundary of
    POLY)
```

```
  (SOMELINESEG POLY (FUNCTION (LAMBDA (PT3 PT4)
```

```
    (CROSSLINES PT1 PT2 PT3 PT4))
```

[189]

```
(SOMELINESEG
```

```
  [LAMBDA (SOMELINESEGX SOMELINESEGFN)
```

```
    (* This is an analogue of SOME that treats a list of
    points (coord pairs) as a list of line segments and
    returns T if SOMELINESEGFN is satisfied by one of
    the line segments. SOMELINESEGFN must be a function
    of two variables for the two points of the line
    segment)
```

```
  (PROG ((SOMELINESEGPT1 (SOMELINESEGX:1)))
```

```
    (if [SOME SOMELINESEGX::1 (FUNCTION (LAMBDA (SOMELINESEGPT2)
```

```

      (PROG1 (APPLY* SOMELINESEGFN SOMELINESEGPT1
                SOMELINESEGPT2)
                SOMELINESEGPT1_SOMELINESEGPT2]
then (RETURN T))

```

[190]

```

(TRACKINPOLY
 [LAMBDA (TRACK POLY)
          (* Determines if a track
            intersects a polygon)
  (SOMELINESEG TRACK (FUNCTION (LAMBDA (TRACKPT1 TRACKPT2)
                                     (LINPOLY TRACKPT1 TRACKPT2 POLY))

```

[191]

```

(CROSSLINES
 [LAMBDA (A B P Q)

  (* The lines AB and PQ cross iff A and B are on
  opposite sides of PQ and P and Q are on opposite
  sides of AB)

```

```

  (AND (OPSIDES A B P Q)
        (OPSIDES P Q A B))

```

[192]

```

(OPSIDES
 [LAMBDA (A B P Q)
          (* Tests if A and B are
            on opposite sides of PQ)
  (ROTSENSE A P Q)=(ROTSENSE Q P B))

```

[193]

```

(ROTSENSE
 [LAMBDA (A B C)
          (* edited:
            "30-Jul-79 10:54")
            (* Tests if the minimal
            rotation from BA to BC
            is clockwise)

  (PROG [(ANGLE ((DIRECTION B:1 B:2 C:1 C:2)
                  -(DIRECTION B:1 B:2 A:1 A:2)
                (RETURN (if ANGLE LT -180.0
                            then T
                            elseif ANGLE GT 180.0
                            then NIL
                            elseif (MINUSP ANGLE)
                            then NIL
                            else T))

```

[194]

```

(SUBTEND
 [LAMBDA (LAT1 LON1 LAT2 LON2)

  (* Gives the angle at the center of the earth

```

subtended by the two points)

```
(COND
  ((EQP LON1 LON2)
   (ABS (FDIFFERENCE LAT1 LAT2)))
  (T (PROG [(C1 (COS (FPLUS LAT1 LAT2)))
            (C2 (COS (FDIFFERENCE LAT1 LAT2)))
            (C3 (COS (FDIFFERENCE LON1 LON2)
                    (RETURN (ARCCOS (FQUOTIENT (FPLUS (FTIMES (FPLUS C1 C2)
                                                            C3)
                                                            C2
                                                            (FMINUS C1))
                    2.0))])])])
```

[195]

```
(LANERANGE
  [LAMBDA (ALAT ALON BLAT BLON CLAT CLON)
    (PROG ((CAT (COS ALAT))
           (SAT (SIN ALAT))
           (CAN (COS ALON))
           (SAN (SIN ALON))
           (CBT (COS BLAT))
           (SBT (SIN BLAT))
           (CBN (COS BLON))
           (SBN (SIN BLON))
           (CCT (COS CLAT))
           (SCT (SIN CLAT))
           (CCN (COS CLON))
           (SCN (SIN CLON))
           (A1 0.0)
           (A2 0.0)
           (B1 0.0)
           (B2 0.0)
           (C1 2.535301E30)
           (C2 0.0)
           A3 B3 C3)
      (SETN A1 (FTIMES CAT CAN))
      (SETN B1 (FTIMES CBT CBN))
      (SETN C1 (FTIMES CCT CCN))
      (SETN A2 (FTIMES CAT SAN))
      (SETN B2 (FTIMES CBT SBN))
      (SETN C2 (FTIMES CCT SCN))
      (SETQ A3 SAT)
      (SETQ B3 SBT)
      (SETQ C3 SCT)
      (RETURN
        (FTIMES
          60.0
          (ABS
            (FDIFFERENCE
              90.0
              (ARCCOS
                (FQUOTIENT
                  [FPLUS [FDIFFERENCE (FTIMES A1
                                      (FDIFFERENCE
```

```

(FTIMES B2 C3)
(FTIMES B3 C2)))
(FTIMES A2
(FDIFFERENCE
(FTIMES B1 C3)
(FTIMES B3 C1)
(FTIMES A3 (FDIFFERENCE (FTIMES B1 C2)
(FTIMES B2 C1)
(SIN (SUBTEND ALAT ALON BLAT BLON))

```

[196]

```

(WITHINR
[NLAMBDA L

```

```

(* NOBIND
"14-Nov-78 19:25")

```

```

(NCONC WITHINR FNS L)
(MAKEFILE (QUOTE WITHINR.LSP))

```

[197]

```

(CROSSPATHS
[LAMBDA (S1 S2 T1 T2)

```

```

(* edited:
"26-Jul-79 12:11")

```

```

(* Tests if path from sightings S1 to S2 crosses
that from T1 to T2)

```

```

(PROG ((P1 (CENTROID S1))
(P2 (CENTROID S2))
(Q1 (CENTROID T1))
(Q2 (CENTROID T2)))
(RETURN (CROSSLINES P1 P2 Q1 Q2)))

```

[198]

```

(LOCATION
[LAMBDA (S)

```

```

(* edited:
"24-Jul-79 17:17")

```

```

(CENTROID (GETATT (QUOTE POSITION)
S))

```

[199]

```

(POSS-REPORT
[LAMBDA (S1 S2 PATROL)

```

```

(* edited:
"24-Jul-79 12:08")

```

```

(PROG (SUCCESSFLG PLAT1)
(SETQ PLAT1 (GETATTB (QUOTE SIGHTING)
S1))
[MAPC (GETATTB (QUOTE SOURCE)
PATROL)
(FUNCTION (LAMBDA (SNG)
(PROG NIL
(PROG ((PLAT2 (GETATTB (QUOTE SIGHTING)
SNG)))
(COND

```

```

((NOT (DISSIMILPLAT PLAT1 PLAT2))
 (SETQ SUCCESSFLG T)
 (RETURN SUCCESSFLG])

```

[200]

```

(DISSIMILPLAT
 [LAMBDA (PLAT1 PLAT2)
 (* edited:
 "19-Jul-79 17:28")

```

```

 (PROG (SUCCESSFLG VAL1 VAL2)
 [MAPC SHIPCHARS (FUNCTION (LAMBDA (CHAR)
 (SETQ VAL1 (GETATT CHAR PLAT1))
 (SETQ VAL2 (GETATT CHAR PLAT2))
 (AND VAL1 VAL2 (NOT (EQUAL VAL1 VAL2))
 (SETQ SUCCESSFLG T)
 (RETURN SUCCESSFLG])

```

[201]

```

(WENT-BEFORE
 [LAMBDA (S1 T1 S2 T2 S3 T3 S4 T4)
 (* edited:
 "30-Jul-79 10:56")
 (* Tests if could have
 gotten from S1 to S2
 before patrol
 overflight)

```

```

 (PROG ((L1 (CENTROID S1))
 (L2 (CENTROID S2))
 (L3 (CENTROID S3))
 (L4 (CENTROID S4))
 THETA PHI VM1 VM2 VP1 VP2 PSI INITDIST FINDIST MINDIST
 MINTIME P0 P4)
 (if T2 lt T3
 then
 (* Got to M2 before
 patrol arrived at P1)

```

```

 (RETURN T))
 (PHI_ (DIRECTION L1:1 L1:2 L2:1 L2:2))
 (VM1_MAXSHIPSPEED*(COS PHI)/60)
 (VM2_MAXSHIPSPEED*(SIN PHI)/60)
 (VP1_ (L3:1-L4:1)/(T3-T4))
 (VP2_ (L3:2-L4:2)/(T3-T4))
 (P0_ <L3:1+(T1-T3)*VP1 L3:2+(T1-T3)*VP2>)
 (P4_ <L3:1+(T2-T3)*VP1 L3:2+(T2-T3)*VP2>)
 (* Projected positions
 of patrol)

```

```

 (PSI_ (ARCTAN (VM2-VP2)/(VM1-VP1)))
 (THETA_ (ABS PSI-(DIRECTION L1:1 L1:2 P0:1 P0:2)))
 (if THETA gt 180
 then THETA_ (360-THETA))
 (INITDIST_ (DISTANCE L1:1 L1:2 P0:1 P0:2))
 (FINDIST_ (DISTANCE L2:1 L2:2 P4:1 P4:2))
 (if THETA gt 90 and INITDIST gt PATROLRANGE
 then (RETURN T))
 (MINDIST_ INITDIST*(SIN THETA))
 (MINTIME_ (60*INITDIST*(COS THETA))/MAXSHIPSPEED+T1)
 (if MINDIST gt PATROLRANGE
 then (RETURN T))

```

```
elseif MINTIME gt T2 and FINDIST gt PATROLRANGE
  then (RETURN T)
```

[202]

```
(WENT-AFTER
[LAMBDA (S1 T1 S2 T2 S3 T3 S4 T4)
```

```
(* edited:
"30-Jul-79 10:55")
(* Tests if could have
gotten from S1 to S2
after patrol overflight)
```

```
(PROG ((L1 (CENTROID S1))
(L2 (CENTROID S2))
(L3 (CENTROID S3))
(L4 (CENTROID S4))
THETA PHI VM1 VM2 VP1 VP2 PSI INITDIST ENDDIST MINDIST
MINTIME P0 P4)
```

```
(if T1 gt T4
  then
```

```
(* Got to M1 after
patrol arrived at P2)
```

```
(RETURN T))
```

```
(PHI_(DIRECTION L1:1 L1:2 L2:1 L2:2))
(VM1_MAXSHIPSPEED*(COS PHI)/60)
(VM2_MAXSHIPSPEED*(SIN PHI)/60)
(VP1_(L3:1-L4:1)/(T3-T4))
(VP2_(L3:2-L4:2)/(T3-T4))
(P0_ <L3:1+(T1-T3)*VP1 L3:2+(T1-T3)*VP2>)
(P4_ <L3:1+(T2-T3)*VP1 L3:2+(T2-T3)*VP2>)
```

```
(* Projected positions
of patrol)
```

```
(PSI_(ARCTAN (VM2-VP2)/(VM1-VP1)))
(THETA_(ABS PSI-(DIRECTION P4:1 P4:2 L2:1 L2:2)))
(if THETA gt 180
  then THETA_(360-THETA))
(INITDIST_(DISTANCE L1:1 L1:2 P0:1 P0:2))
(ENDDIST_(DISTANCE L2:1 L2:2 P4:1 P4:2))
(if THETA gt 90 and ENDDIST gt PATROLRANGE
  then (RETURN T))
(MINDIST_ENDDIST*(SIN THETA))
(MINTIME_(-60*ENDDIST*(COS THETA))/MAXSHIPSPEED+T2)
(if MINDIST gt PATROLRANGE
  then (RETURN T)
  elseif MINTIME lt T1 and INITDIST gt PATROLRANGE
  then (RETURN T))
```

[203]

```
(LOC-TIME
[LAMBDA (S)
```

```
(* edited:
"24-Jul-79 17:51")
```

```
(NCONC1 (CENTROID (GETATT (QUOTE POSITION)
S))
(GETATT (QUOTE TOS)
S))
```

[204]

(SWR
[LAMBDA (LT1 T1 LT2 T2)

(* edited:
"26-Jul-79 12:25")

(* Tests if sighting S1 is simply-within-reach of
S2, ie. by travelling straight ahead with max ship
speed)

(PROG ((L1 (CENTROID LT1))
(L2 (CENTROID LT2)))
(RETURN (LESSP (SPEEDM T1 T2 (DISTANCE (CAR L1)
(CADR L1)
(CAR L2)
(CADR L2)))

MAXSHIPSPEED])

[205]

(SPEEDM
[LAMBDA (T1 T2 DIST)

(* edited:
" 8-Aug-79 09:09")

(ABS (SPEEDAUX (FQUOTIENT T1 60)
(FQUOTIENT T2 60)
DIST])

[206]

(GRAZE
[LAMBDA (S1 S2 T1 T2)

(* edited:
" 6-Aug-79 20:03")

(* Given two sightings of each of two platforms,
graze returns true if the paths of the platforms are
within the patrolrange. Time is not considered.)

(PROG ((POS1 (CENTROID S1))
(POS2 (CENTROID S2))
(POS3 (CENTROID T1))
(POS4 (CENTROID T2)))
(RETURN (OR (LESSP (DISTOLINE (CAR POS1)
(CADR POS1)
(CAR POS3)
(CADR POS3)
(CAR POS4)
(CADR POS4))
PATROLRANGE)
(LESSP (DISTOLINE (CAR POS2)
(CADR POS2)
(CAR POS3)
(CADR POS3)
(CAR POS4)
(CADR POS4))
PATROLRANGE]))

[207]

(SUCCESSOR
[LAMBDA (SITE)

(* edited:
" 6-Aug-79 20:07")

(* Given a sighting node this function returns the
next sighting in time or nil if there is no
successor)

```
(PROG (SUCC TOSX TOSSUC TOSSITE PLAT)
  (SETQ PLAT (GETATTB (QUOTE SIGHTING)
                      SITE))
  (SETQ TOSSITE (GETATT (QUOTE TOS)
                        SITE))
  [for X in (RETRIEVES (QUOTE SIGHTING)
                      PLAT
                      (QUOTE *))
    3)
  do (PROG NIL
    (SETQ TOSX (GETATT (QUOTE TOS)
                       X))
    (COND
      ((LESSP TOSSITE TOSX)
       (COND
         ((OR (NULL SUCC)
              (LESSP TOSX TOSSUC))
          (SETQ SUCC X)
          (SETQ TOSSUC TOSX)
          (RETURN SUCC))
```

[208]

(PREDECESSOR
[LAMBDA (SITE)

(* edited:
" 6-Aug-79 20:09")

(* Given a sighting node this function returns the
previous sighting in time, or nil if there was no
previous sighting.)

```
(PROG (PRED TOSX TOSPRED TOSSITE PLAT)
  (SETQ PLAT (GETATTB (QUOTE SIGHTING)
                      SITE))
  (SETQ TOSSITE (GETATT (QUOTE TOS)
                        SITE))
  [for X in (RETRIEVES (QUOTE SIGHTING)
                      PLAT
                      (QUOTE *))
    3)
  do (PROG NIL
    (SETQ TOSX (GETATT (QUOTE TOS)
                       X))
    (COND
```

```

      ((LESSP TOSX TOSSITE)
      (COND
        ((OR (NULL PRED)
              (LESSP TOSPREP TOSX))
         (SETQ PRED X)
         (SETQ TOSPREP TOSX)
        (RETURN PRED]))

```

[209]

```

(DIRECTION
 [LAMBDA (LAT1 LON1 LAT2 LON2)

```

```

(* edited:
 " 7-Aug-79 17:40")

```

```

  (PROG ((PSI 0.0)
         (LONDIF 0.0)
         (BEARSIN 0.0)
         (BEARANGLE 0.0))
    (SETN PSI (SUBTEND LAT1 LON1 LAT2 LON2))
    (SETN LONDIF (FDIFFERENCE LON2 LON1))
    (COND
      ((EQP LAT1 90.0)
       (RETURN 180.0))
      ((EQP LAT1 -90.0)
       (RETURN 0.0))
      (SETN BEARSIN (FQUOTIENT (FTIMES (COS LAT2)
                                         (SIN LONDIF))
                               (SIN PSI)))
    (COND
      ((FGTP BEARSIN 1.0)
       (SETN BEARSIN 1.0))
    (COND
      ((LESSP BEARSIN -1.0)
       (SETN BEARSIN -1.0))
      (SETN BEARANGLE (ARCSIN BEARSIN))
    (COND
      ((LESSP LAT2 LAT1)
       (SETN BEARANGLE (FDIFFERENCE 180.0 BEARANGLE))
    (COND
      ((MINUSP BEARANGLE)
       (SETN BEARANGLE (FPLUS 360.0 BEARANGLE))
      (RETURN BEARANGLE))

```

[210]

```

(RANGE
 [LAMBDA (SITE)

```

```

(* edited:
 " 6-Aug-79 20:15")

```

```

(* Range accepts a sighting node and computes the
distance from the ownship to the platform sighted.)

```

```

(PROG (POS1 POS2 TIME)
  (SETQ TIME (GETATT (QUOTE TOS)
                     SITE))
  (SETQ POS1 (OWNPOS TIME))
  (SETQ POS2 (CENTROID (GETATT (QUOTE POSITION)

```

```

                                SITE)))
  (RETURN (DISTANCE (CAR POS1)
                    (CADR POS1)
                    (CAR POS2)
                    (CADR POS2)))

```

[211]

```

(COURSE
  [LAMBDA (SITE)

```

```

(* edited:
  " 7-Aug-79 10:53")

```

```

(* Course accepts a sighting node and computes an
estimated course. To do this the closer
(in time) of the predecessor or successor is used.)

```

```

(PROG (PRED SUC TPRED TSUC PPRED PSUC POS TIME)
  (SETQ TIME (GETATT (QUOTE TOS)
                    SITE))
  (SETQ POS (CENTROID (GETATT (QUOTE POSITION)
                              SITE)))
  (SETQ PRED (PREDECESSOR SITE))
  (SETQ SUC (SUCCESSOR SITE))
  [COND
    (SUC (SETQ TSUC (GETATT (QUOTE TOS)
                            SUC))
          (SETQ PSUC (CENTROID (GETATT (QUOTE POSITION)
                                        SUC)
                              SUC)
          [COND
            (PRED (SETQ TPRED (GETATT (QUOTE TOS)
                                      PRED))
                  (SETQ PPRED (CENTROID (GETATT (QUOTE POSITION)
                                                PRED)
                                      PRED)
            (COND
              ((AND (NULL PRED)
                    (NULL SUC)
                     (RETURN)))
              [(NULL PRED)
               (RETURN (DIRECTION (CAR POS)
                                  (CADR POS)
                                  (CAR PSUC)
                                  (CADR PSUC)
              [(NULL SUC)
               (RETURN (DIRECTION (CAR PPRED)
                                  (CADR PPRED)
                                  (CAR POS)
                                  (CADR POS)
              [(LESSP (FDIFFERENCE TIME TPRED)
                      (FDIFFERENCE TSUC TIME))
               (RETURN (DIRECTION (CAR PPRED)
                                  (CADR PPRED)
                                  (CAR POS)
                                  (CADR POS)
              (T (RETURN (DIRECTION (CAR POS)
                                    (CADR POS)
                                    (CAR PSUC)

```

(CADR PSUC1)

[212]

```
(SPEEDAUX
  [LAMBDA (T1 T2 DIST)
    (FQUOTIENT DIST (FDIFFERENCE T2 T1))
```

```
(* edited:
  "30-Jul-79 18:59")
```

[213]

```
(COURSEFROM
  [LAMBDA (POS1 POS2)
    (SETQ POS1 (CENTROID POS1))
    (SETQ POS2 (CENTROID POS2))
    (DIRECTION (CAR POS1)
      (CADR POS1)
      (CAR POS2)
      (CADR POS2))
```

```
(* edited:
  " 7-Aug-79 17:36")
```

[214]

```
(SPEEDFROM
  [LAMBDA (POS1 T1 POS2 T2)
    (SETQ POS1 (CENTROID POS1))
    (SETQ POS2 (CENTROID POS2))
    (SPEEDM T1 T2 (DISTANCE (CAR POS1)
      (CADR POS1)
      (CAR POS2)
      (CADR POS2))
```

```
(* edited:
  " 7-Aug-79 17:54")
```

```
)
(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVAR
```

```
(ADDTOVAR NLAMA WITHINR)
```

```
(ADDTOVAR NLAML )
```

```
(ADDTOVAR LAMA )
```

```
)
(DECLARE: DONTCOPY
```

```
(FILEMAP (NIL (2459 25278 (SAME-AS 2471 . 2596) (
ROUGHLY-THE-SAME-SPEED-AS 2600 . 3038) (ROUGHLY-THE-SAME-COURSE-AS 3042
. 3424) (IN-LANE 3428 . 3958) (INSIDE 3962 . 4322) (GETATTB 4326 . 4652)
(GREATER-THAN 4656 . 4794) (LESS-THAN 4798 . 4933) (BEARING 4937 . 5532
) (SPEED 5536 . 7000) (INTERIOR 7004 . 8061) (DISTANCE 8065 . 8154) (
DISTOLINE 8158 . 8943) (INLANE 8947 . 9441) (LINPOLY 9445 . 9639) (
CROSSBOUNDARY 9643 . 9892) (SOMELINESEG 9896 . 10516) (TRACKINPOLY 10520
. 10739) (CROSSLINES 10743 . 10962) (OPSIDES 10966 . 11120) (ROTSENSE
11124 . 11616) (SUBTEND 11620 . 12070) (LANERANGE 12074 . 13253) (
WITHINR 13257 . 13413) (CROSSPATHS 13417 . 13807) (LOCATION 13811 .
13972) (POSS-REPORT 13976 . 14572) (DISSIMILPLAT 14576 . 14972) (
WENT-BEFORE 14976 . 16560) (WENT-AFTER 16564 . 18136) (LOC-TIME 18140 .
18348) (SWR 18352 . 18841) (SPEEDM 18845 . 19038) (GRAZE 19042 . 19818)
(SUCCESSOR 19822 . 20659) (PREDECESSOR 20663 . 21518) (DIRECTION 21522 .
22474) (RANGE 22478 . 23172) (COURSE 23176 . 24559) (SPEEDAUX 24563 .
```

24701) (COURSEFROM 24705 . 24972) (SPEEDFROM 24976 . 25275))))
STOP

(FILECREATED " 6-Aug-79 11:01:50" <DKIBLER>PLAT.LSP.48 10184

changes to: PLATFNS GETATT ESTIMATE NEAREST ONEPOINT CENTROID
GETPOINT FIXLONG

previous date: "24-Jul-79 18:00:26" <DKIBLER>PLAT.LSP.47)

(PRETTYCOMPRINT PLATCOMS)

(RPAQQ PLATCOMS ((FNS * PLATFNS)))

(RPAQQ PLATFNS (GETATT ESTIMATE NEAREST ONEPOINT PLATPOS PREDICTPOS
SPAN AUXINTERPOL CENTROID GETPOINT FIXLONG))

(DEFINEQ

[215]

(GETATT
[LAMBDA (REL NAME)

(* edited:
" 6-Aug-79 10:24")

(* This function accepts a two-argument relation
name together with its first argument and returns
the first instance of the second argument which
satisfies the relation. It is useful when the
relation is infact a function.)

(PROG [(SPEC (QUOTE (NIL NIL *)
(RPLACA SPEC REL)
(RPLACA (CDR SPEC)
NAME)
(RETURN (CADDR (GETUPLE (CAR (STRIPSTREAM (GETSH SPEC))

[216]

(ESTIMATE
[LAMBDA (SITE1 SITE2 GAP)

(* edited:
" 6-Aug-79 10:33")

(* This function accepts two sightings and a
normalized time factor and returns an estimated
position. The estimated position may be in the
future or past, and will be a polygon if either
position of a sighting is a polygon.
The normalize time factor is the desired time minus
the time of sighting1 divided by the difference in
the time of the sightings.)

(PROG1 [MAPCAR (SPAN (GETATT (QUOTE POSITION)
SITE1)
(GETATT (QUOTE POSITION)
SITE2))
(FUNCTION (LAMBDA (X)

```

(AUXINTERPOL (CAR X)
              (CADR X)
              GAP]
(AND EXPLAINFLAG (PRIN1 "Estimated from the sightings ")
  (PRIN1 SITE1)
  (PRIN1 " and ")
  (PRIN1 SITE2)
  (TERPRI))

```

[217]

```

(NEAREST
 [LAMBDA (PT LST)

```

```

(* edited:
 " 6-Aug-79 10:40")

```

```

(* Given a point P and a list of points L, this
function returns the point of L nearest to P.
Each point is a latitude-longitude pair.)

```

```

(PROG ((ANS (CAR LST))
      Y TEMP)
  (SETQ TEMP (DISTANCE (CAR PT)
                      (CADR PT)
                      (CAR ANS)
                      (CADR ANS)))
  [for X in (CDR LST) do (COND
                        ((FLESSP (SETQ Y
                                  (DISTANCE (CAR PT)
                                             (CADR PT)
                                             (CAR X)
                                             (CADR X)))
                                TEMP)
                          (SETQ ANS X)
                          (SETQ TEMP Y)
                        )
  (RETURN ANS))

```

[218]

```

(ONEPOINT
 [LAMBDA (NODE GAP)

```

```

(* edited:
 " 6-Aug-79 10:44")

```

```

(* Given a single sighting and a time relative to
that sighting, this function generates an estimated
position which will be a polygon.)

```

```

(PROG ((X (FTIMES .5 GAP))
      (POS (GETATT (QUOTE POSITION)
                  NODE))
      LAT LONG)
  (AND EXPLAINFLAG (PRIN1 "The only sighting node is ")
    (PRIN1 NODE)
    (PRIN1
      " and no course was known. Hence the polygon is large.")
    (TERPRI))

```

```
(COND
  [(NULL (CDR POS))
   (SETQ LAT (CAAR POS))
   (SETQ LONG (CADAR POS))
   (SETQ X (FTIMES .5 GAP))
```

(* .5 is the approximate speed in degrees of a vessel. (60 knots=1 degree per hour))

```
(RETURN (LIST (LIST (FDIFFERENCE LAT X)
                     (FDIFFERENCE LONG X))
              (LIST (FPLUS LAT X)
                     (FDIFFERENCE LONG X))
              (LIST (FPLUS LAT X)
                     (FPLUS LONG X))
              (LIST (FDIFFERENCE LAT X)
                     (FPLUS LONG X)
              (T (RETURN (MAPCAR (SPAN (LIST (CENTROID POS))
                                         POS)
                                (FUNCTION (LAMBDA (Y)
                                             (AUXINTERPOL (CAR Y)
                                                            (CADR Y)
                                                            X))
```

[219]

```
(PLATPOS
  [LAMBDA (PLAT TIME)
```

(* edited:
"11-Jul-79 13:10")

(* Given a platform and a time this function returns the latitude and longitude if an appropriate sighting has been made. If there are bounding sightings, a position is estimated by interpolation. If there are no bounding sightings, a polygon is computed by extrapolation and returned.)

```
(PROG (X Y)
  (SETQ X (MAPCAR (RETRIEVER (LIST (QUOTE SIGHTING)
                                   PLAT
                                   (QUOTE *)))
                 (FUNCTION CDADR)))
  (COND
    [[SETQ Y (SUBSET X (FUNCTION (LAMBDA (Z)
                                   (EQUAL (GETATT (QUOTE TOS)
                                                    Z)
                                   TIME)
    (AND EXPLAINFLAG (PRIN1
                      "We have a sighting of the platform.")
    (TERPRI))
    (RETURN (GETATT (QUOTE POSITION)
                    (CAR Y)
    (X (RETURN (PREDICTPOS X TIME)))
    (EXPLAINFLAG (PRIN1 "No sighting of platform exists."))
```

(TERPRI))

[220]

(PREDICTPOS

[LAMBDA (NODELIST TIME)

(* edited:
"11-Jul-79 17:54")

(* This function distributes the task of computing
an approximate position depending on the number and
type of sightings.)

```

(PROG (LB UB LBT UBT LB2 UB2 LBT2 UB2)
  [MAPC NODELIST (FUNCTION (LAMBDA (X)
    (PROG (XT)
      (SETQ XT (GETATT (QUOTE TOS)
                       X))
      (COND
        [(FLESSP XT TIME)
         (COND
           ((OR (NULL LB)
                (FLESSP LBT XT))
            (SETQ LB2 LB)
            (SETQ LBT2 LBT)
            (SETQ LB X)
            (SETQ LBT XT))
           ((OR (NULL LBT2)
                (FLESSP LBT2 XT))
            (SETQ LBT2 XT)
            (SETQ LB2 X)
            ((COND
              ((OR (NULL UB)
                   (FLESSP XT UBT))
               (SETQ UB2 UB)
               (SETQ UBT2 UBT)
               (SETQ UB X)
               (SETQ UBT XT))
              ((OR (NULL UBT2)
                   (FLESSP XT UBT2))
               (SETQ UBT2 XT)
               (SETQ UB2 X))))
            (SETQ UB X)
            (SETQ UBT XT)
          (RETURN (COND
            [(AND UB LB)
             (ESTIMATE LB UB (FQUOTIENT (FDIFFERENCE TIME LBT)
                                         (FDIFFERENCE UBT LBT)
            [UB2 (ESTIMATE UB UB2 (FQUOTIENT (FDIFFERENCE
                                              TIME UBT)
                                              (FDIFFERENCE
                                              UBT2 UBT)
            [LB2 (ESTIMATE LB LB2 (FQUOTIENT (FDIFFERENCE
                                              TIME LBT)
                                              (FDIFFERENCE
                                              LBT LBT2)
            (UB (ONEPOINT UB (FDIFFERENCE UBT TIME)))

```

(LB (ONEPOINT LB (FDIFFERENCE TIME LBT))

[221]

(SPAN

[LAMBDA (L1 L2)

(* edited:

"11-Jul-79 16:31")

(* This function takes two polygons
(possibly degenerate) and generates an approximation
to the span of this polygons.)

(COND

[(IGREATERP (LENGTH L1)

(LENGTH L2))

(MAPCAR L1 (FUNCTION (LAMBDA (X)

(LIST X (NEAREST X L2)

(T (MAPCAR L2 (FUNCTION (LAMBDA (X)

(LIST (NEAREST X L1)

X))

[222]

(AUXINTERPOL

[LAMBDA (PT1 PT2 DELTA)

(* edited:

"16-Jul-79 18:33")

(LIST [FPLUS (CAR PT1)

(FTIMES DELTA (FDIFFERENCE (CAR PT2)

(CAR PT1)

(FIXLONG (FPLUS (CADR PT1)

(FTIMES DELTA (FIXLONG (FDIFFERENCE

(CADR PT2)

(CADR PT1))

[223]

(CENTROID

[LAMBDA (VERTEXLIST)

(* edited:

" 6-Aug-79 10:46")

(* Given a list of points, which are
latitude-longitude pairs, this function returns the
centroid of those points.)

(PROG ((C1 (CAAR VERTEXLIST))

(C2 (CADAR VERTEXLIST))

(I 1))

[COND

((NULL (CDR VERTEXLIST))

(RETURN (CAR VERTEXLIST)

LOOP(COND

[(NULL (CDR VERTEXLIST))

(RETURN (LIST (FQUOTIENT C1 I)

(FQUOTIENT C2 I)

(T (SETQ I (ADD1 I))

```

(SETQ VERTEXLIST (CDR VERTEXLIST))
(SETQ C1 (FPLUS C1 (CAAR VERTEXLIST)))
(SETQ C2 (FPLUS C2 (CADAR VERTEXLIST)))
(GO LOOP)

```

[224]

```

(GETPOINT
 [LAMBDA (POS BEAR RANGE)
  (CLISP: FLOATING)

```

```

(* edited:
 " 6-Aug-79 10:58")

```

```

(* This function returns the new position reached by
traveling from the given position
(a latitude-longitude pair) at the given bearing for
the given range.)

```

```

(PROG ((LAT (POS:1))
       (LONG (POS:2))
       (PSI (RANGE/60))
       NEWLAT NEWLONG TMP TMP2 SINLAT COSPSI COSLAT SINPSI COSBEAR
       COSNEWLAT)
  (SINLAT_(SIN LAT))
  (COSPSI_(COS PSI))
  (COSLAT_(COS LAT))
  (SINPSI_(SIN PSI))
  (COSBEAR_(COS BEAR))
  (NEWLAT_(ARCSIN SINLAT*COSPSI+COSLAT*SINPSI*COSBEAR))
  (if (EQUAL 90 (ABS NEWLAT))
      then (RETURN <NEWLAT 0>))
  (COSNEWLAT_(COS NEWLAT))
  (TMP_SINPSI*(SIN BEAR)/COSNEWLAT)
  [TMP2_(ARCCOS (MAX -1 (MIN 1 (
                                COSLAT*COSPSI-COSBEAR*SINLAT*SINPSI)
                                /COSNEWLAT)
  (NEWLONG_LONG+(if TMP gt 0
                    then TMP2
                    else (-TMP2)))
  (NEWLONG_(FIXLONG NEWLONG))
  (RETURN <NEWLAT NEWLONG>)]

```

[225]

```

(FIXLONG
 [LAMBDA (X)

```

```

(* edited:
 " 6-Aug-79 11:01")

```

```

(* Given a longitude whose absolute value is less
than 360, this function will return a longitude in
the proper range.)

```

```

(COND
 ((FLESSP 180 X)
  (FDIFFERENCE X 360))
 ((FLESSP X -180)

```

(FPLUS X 360))
(T X))
)

(DECLARE: DONTCOPY

(FILEMAP (NIL (422 10160 (GETATT 434 . 1065) (ESTIMATE 1069 . 2031) (NEAREST 2035 . 2810) (ONEPOINT 2814 . 4086) (PLATPOS 4090 . 5041) (PREDICTPOS 5045 . 6582) (SPAN 6586 . 7128) (AUXINTERPOL 7132 . 7466) (CENTROID 7470 . 8252) (GETPOINT 8256 . 9714) (FIXLONG 9718 . 10157))))))
STOP

(FILECREATED "21-Aug-79 12:09:01" <PMORRIS>QH.LSP.72 8186

changes to: QHASK

previous date: " 1-Aug-79 20:28:51" <PMORRIS>QH.LSP.71)

(PRETTYCOMPRINT QHCOMS)

(RPAQQ QHCOMS [(MACROS QHGET QHPUT)

(FNS * QHFNS)

(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
COMPILEVAR (ADDVAR (NLAMA QHTAKE PQ)
(NLAML)
(LAMA])

(DECLARE: EVAL@COMPILE

(PUTPROPS QHGET MACRO [(LOC OFF)

(GETHASH (VAG (IPLUS (ITIMES LOC 132)
OFF 3])

(PUTPROPS QHPUT MACRO ((LOC OFF VAL)

(PUTHASH (VAG (IPLUS (ITIMES LOC 132)
OFF 3))

VAL)))

)

(RPAQQ QHFNS (PQ QHCLEAR QHMAKE QHLIST QHASK BEEP QHTAKE QHFOLLOW
QHPREP QHSHOW))

(DEFINEQ

[226]

(PQ

[NLAMBDA L

(* edited:
" 1-Aug-79 13:26")

(PROG ((SYSPRETTYFLG T))

(SHOWPRINT (GETPROP (CAR L)

(QUOTE QHPRODS))

[227]

(QHCLEAR

[LAMBDA NIL

(* edited:
"15-Jun-79 18:44")

(SETQ QUERYHASHPTR 0)

(CLRHASH)

[228]

(QHMAKE

[LAMBDA (QHMAKEX QHMAKEY SHOWFLG)

(* edited:
"27-Jul-79 18:33")

(PROG ((PTR 0)

NEWPTR CHARCODE)

(COND

((LISTP QHMAKEX)

```

[MAPC QHMAKEX (FUNCTION (LAMBDA (X)
  (QHMAKE X QHMAKEY SHOWFLG]
  (RETURN))
((EQ (NTHCHAR QHMAKEX 1)
  (QUOTE !))
  (COND
    (SHOWFLG (PRIN1 (COND
      ((MEMB (NTHCHAR QHMAKEX 2)
        (QUOTE (A E I O U)))
        "an ")
      (T "a ")))
      (PRIN1 (SUBSTRING QHMAKEX 2))
      (TERPRI))
    (T (QHMAKE (EVAL (MKATOM (SUBSTRING QHMAKEX 2)))
      QHMAKEY)))
  (RETURN)))
(COND
  (SHOWFLG (PRINT QHMAKEX)
    (RETURN)))
[RPQTQ (NCHARS QHMAKEX)
  (PROGN [SETQ CHARCODE (CHCON1 (NTHCHAR QHMAKEX
    (IMINUS RPTN]
    (SETQ NEWPTR (QHGET PTR CHARCODE))
    (COND
      (NEWPTR (SETQ PTR NEWPTR))
      (T (QHPUT PTR CHARCODE (SETQ QUERYHASHPTR
        (ADD1 QUERYHASHPTR)))
        (QHPUT PTR 0 (COND
          ((QHGET PTR 0)
            T)
          (T CHARCODE)))
        (SETQ PTR QUERYHASHPTR]
    (QHPUT PTR -1 QHMAKEX)
    (QHPUT PTR -2 QHMAKEY])

```

[229]

```

(QHLIST
  [LAMBDA (PTR)
    (* edited:
      "20-Jun-79 15:32")
    (PROG (ITEM)
      (COND
        (PTR (SETQ ITEM (QHGET PTR -1))
          [COND
            (ITEM (COND
              ((EQ (NTHCHAR ITEM -1)
                (QUOTE $))
                (* Escape)
              (PRIN1 (SUBSTRING ITEM 1 -2))
              (PRIN1 "<anything>")
              (TERPRI))
              (T (PRINT ITEM]
            (COND
              ((QHGET PTR 0)
                (RPTQ 129 (QHLIST (QHGET PTR (IDIFFERENCE 130 RPTN]

```

[230]

(QHASK

[LAMBDA (INBUF)

(* edited:

"21-Aug-79 12:08")

(RESETLST

(RESETSAVE (CONTROL T))

(RESETSAVE (ECHOMODE))

(RESETSAVE (RAISE T))

(RESETSAVE ([LAMBDA (X)

(ECHOCONTROL 8 X]

(QUOTE REAL)))

(PROG (PTR BUFPTR NEWPTR CHAR ITEM CODE)

(COND

((EQ (QHGET 0 0)

63)

(* Nothing except query)

(RETURN)))

ENTRY

[COND

((EQ (PEEKC)

(QUOTE &))

(TERPRI)

(RETURN (CONS (PRIN1 (READC]

(SETQ PTR 0)

(SETQ BUFPTR (CONS))

LOOP[OR (QHGET PTR 0)

(RETURN (CONS (QHGET PTR -1)

(QHGET PTR -2]

(SETQ CHAR (READC))

(SETQ CODE (CHCON1 CHAR))

(SETQ NEWPTR (QHGET PTR CODE))

(COND

(NEWPTR (PRIN1 CHAR)

(TCONC BUFPTR CHAR)

(SETQ PTR NEWPTR))

[(MEMB CODE (GETSYNTAX (QUOTE SEPRCHAR)))

(SETQ ITEM (QHGET PTR -1))

(COND

[ITEM (PRIN1 CHAR)

(RETURN (CONS ITEM (QHGET PTR -2]

(T (BEEP]

((EQ CODE 63)

(* Query)

(TERPRI)

(PRIN1 "one of:")

(TERPRI)

(QHLIST PTR)

(TERPRI)

(MAPRINT INBUF)

(PRIN1 " ")

(MAPC (CAR BUFPTR)

(FUNCTION PRIN1)))

((EQ CODE 27)

(* Escape)

(PROG (NUM)

(SETQ NUM (QHGET PTR 0))

```

      INLOOP
      (COND
        ((OR (NOT (NUMBERP NUM))
              (QHGET PTR -1))
          (COND
            (NUM (BEEP)))
          (RETURN)))
      (TCONC BUFPTR (PRIN1 (CHARACTER NUM)))
      (SETQ PTR (QHGET PTR NUM))
      (SETQ NUM (QHGET PTR 0))
      (GO INLOOP))
((EQ CODE 127)
  (* Rubout)
  (TERPRI)
  (MAPRINT INBUF)
  (PRIN1 " ")
  (GO ENTRY))
[(AND (SETQ NEWPTR (QHGET PTR 27))
      (SETQ ITEM (QHGET NEWPTR -1)))
 (RETURN (CONS [MKATOM (CONCAT (OR (SUBSTRING ITEM 1 -2)
                                   " "))
               (PRIN1 CHAR)
               (RESETFORM (ECHOMODE
                           T)
                           (READ]
               (QHGET NEWPTR -2]
               (T (BEEP)))
               (GO LOOP])

```

[231]

```

(BEEP
 [LAMBDA NIL
  (PRIN1 (CHARACTER 71))
  (* edited:
  "15-Jun-79 13:24")

```

[232]

```

(QHTAKE
 [NLAMBDA L
  (QHFOLLOW (CONS L)
             (CONS))
  (* edited:
  "12-Jul-79 19:25")

```

[233]

```

(QHFOLLOW
 [LAMBDA (LL BUFPTR QHMATCH)
  (PROG (L X QHVAL ALIST)
    START
      (COND
        ((CDR LL)
         (SETQ QHMATCH (QHFOLLOW (CDR LL)
                                   BUFPTR QHMATCH])
         (SETQ L (CAR LL))
        LOOP(COND
  (* edited:
  " 1-Aug-79 19:37")

```

```

      ((NULL L)
       (RETURN QHMATCH)))                                (* Default)
    (SETQ X (CAR L))
    (COND
      ((EQ X (QUOTE :))
       (RETURN (EVALA (CADR L)
                       ALIST)))
      ((EQ (CAR X)
            (QUOTE =))
       (SETQ ALIST (CONS (CONS (CADR X)
                                QHMATCH)
                          ALIST))
       (SETQ L (CDR L))
       (GO LOOP))
      ((STRINGP X)
       [TCONC BUFPTR (PRIN1 (COND
                             ((EQ (NTHCHAR X 1)
                                   (QUOTE -))
                              (CONCAT (CHARACTER 8)
                                       (SUBSTRING X 2)))
                             (T X]

      (PRIN1 " ")
      (SETQ L (CDR L))
      (GO LOOP)))
    (QHCLEAR)
    (QHPREP (CAR L)
             (CDR L))
    (QHMAKE (QUOTE ?))
  QUERY
    (SETQ QHVAL (QHASK (CAR BUFPTR)))
    (COND
      ((EQ (CAR QHVAL)
            (QUOTE ?))
       (TERPRI)
       (PRIN1 "one of:")
       (TERPRI)
       (QHSHOW L)
       (TERPRI)
       (MAPRINT (CAR BUFPTR))
       (PRIN1 " ")
       (GO QUERY))
      ((EQ (CAR QHVAL)
            (QUOTE &))
       (SETQ QHVAR (READ))
       (CLEARBUF)
       [COND
         ((NOT (LISTP (EVALV QHVAR)))
          (PRIN1 "??"))
         (T (TERPRI)
              (PRIN1 "one of")
              (TERPRI)
              (QHSHOW (CONS (EVAL QHVAR)
                             (CDR QHVAR)))
              (TERPRI)
              (MAPRINT (CAR BUFPTR))
              (PRIN1 " ")
              (GO QUERY))
        (SETQ QHMATCH (CAR QHVAL))

```

```
(SETQ LL (REVERSE (CDR QHVAL)))
(PRIN1 " ")
(TCONC BUFPTR (CAR QHVAL))
(GO START)
```

[234]

```
(QHPREP
[LAMBDA (FOCUS QHLST SHOWFLG STK)
(COND
[(LISTP FOCUS)
(COND
((EQ (CAR FOCUS)
(QUOTE ~))
(QHPREP (CDR FOCUS)
QHLST SHOWFLG STK)
(QHPREP (CAR QHLST)
(CDR QHLST)
SHOWFLG STK))
(T (MAPC FOCUS (FUNCTION (LAMBDA (F)
(QHPREP F QHLST SHOWFLG STK)
[(AND (EQ (NTHCHAR FOCUS 1)
(QUOTE <))
(EQ (NTHCHAR FOCUS -1)
(QUOTE >)))
(MAPC (OR (GETPROP FOCUS (QUOTE QHPRODS))
(HELP "No productions for" FOCUS))
(FUNCTION (LAMBDA (X)
(QHPREP (CAR X)
(CDR X)
SHOWFLG
(CONS QHLST STK)
(T (QHMAKE FOCUS (CONS QHLST STK)
SHOWFLG))
(* edited:
" 1-Aug-79 12:17")
```

[235]

```
(QHSHOW
[LAMBDA (L)
(QHPREP (CAR L)
(CDR L)
T))
)
(DECLARE: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVAR
(ADDTOVAR NLAMA QHTAKE PQ)
(ADDTOVAR NLAML )
(ADDTOVAR LAMA )
)
(DECLARE: DONTCOPY
(FILEMAP (NIL (727 8025 (PQ 739 . 965) (QHCLEAR 969 . 1120) (QHMAKE 1124
. 2309) (QHLIST 2313 . 2883) (QHASK 2887 . 5007) (BEEP 5011 . 5130) (
QHTAKE 5134 . 5268) (QHFOLLOW 5272 . 7119) (QHPREP 7123 . 7851) (QHSHOW
(* edited:
"13-Jul-79 16:23")
```

<PMORRIS>QH.LSP.72

Page 153

7855 . 8022))))
STOP

(FILECREATED "27-Aug-79 21:39:40" <RBECHTAL>RULES..29 16305

changes to: (MATCH-PLAT CONDITIONS)

previous date: "27-Aug-79 18:40:58" <RBECHTAL>RULES..28)

(PRETTYCOMPRINT RULESCOMS)

```
(RPAQQ RULESCOMS ((VARS * RULESVARS)
                  (FNS * RULESFNS)
                  (PROP (CONDITIONS ACTIONS CONF)
                        * PRODUCTIONS)))
```

(RPAQQ RULESVARS (PRODUCTIONS))

```
(RPAQQ PRODUCTIONS (INHERIT NOT-LAST-SIGHTING NOT-FIRST-SIGHTING
                        FIRST-VIEW NOT-KNOWN-COMBATANT REACHABLE
                        SIMPLY-REACHABLE POSS-RPT BLOCKER
                        CREATEDETECT CREATECONTACT CREATEPLAT
                        SMALL-CRAFT9 SMALL-CRAFT6 SMALL-CRAFT5
                        SMALL-CRAFT4 SMALL-CRAFT3 SMALL-CRAFT2
                        SMALL-CRAFT1 ID-LANE INSIDE-A-STORM
                        CLOSE-POPUP DISTANT-POPUP COURSE-CHANGED
                        SPEED-CHANGED FASTER-THAN-A-MERCHANT
                        SLOWER-THAN-A-MERCHANT MATCH-PLAT
                        OUTSIDE-ALL-LANES))
```

```
(RPAQQ RULESFNS (DEFINEPD MAKEPD FANCYPROD LINEREAD))
(DEFINEQ
```

[236]

```
(DEFINEPD
 [LAMBDA NIL
```

```
(* edited:
 "27-Jun-79 17:33")
```

(PROG (PDNAME NEWCON CONDS NEWACT ACT CONFID)

```
(* DEFINEPD provides a "nice" user interface for
production rule definition, by prompting for needed
information.)
```

```
(PRIN1 "NAME? ")
(SETQ PDNAME (CAR (LINEREAD)))
(PRIN1 "CONDITIONS? ")
DPD1(COND
      ((SETQ NEWCON (LINEREAD))
       (SETQ CONDS (APPEND CONDS NEWCON))
       (GO DPD1)))
(PRIN1 "ACTION? ")
DPD2(COND
      ((SETQ NEWACT (LINEREAD))
       (SETQ ACT (APPEND ACT NEWACT))
       (GO DPD2)))
(PRIN1 "CONFIDENCE? ")
```

```
(SETQ CONFID (CAR (LINEREAD)))
(MAKEPD PDNAME CONDS ACT CONFID))
```

[237]

```
(MAKEPD
  [LAMBDA (NAM CO AC TRUST)
```

```
(* edited:
  "27-Jun-79 10:26")
```

```
(PROG NIL
```

```
(* MAKEPD does the actual construction of
  productions. The elements of a production are stored
  on the property list of the production name.)
```

```
(PUTPROP NAM (QUOTE CONDITIONS)
  CO)
(PUTPROP NAM (QUOTE ACTIONS)
  AC)
(PUTPROP NAM (QUOTE CONF)
  TRUST)
(SETQ PRODUCTIONS (CONS NAM PRODUCTIONS))
(RETURN NAM))
```

[238]

```
(FANCYPROD
  [LAMBDA (PRO)
```

```
(* edited:
  "27-Aug-79 17:18")
(* FANCYPROD is a
  prettyprinter for
  productions.)
```

```
(PRIN1 "NAME: ")
(PRIN1 PRO)
(TERPRI)
(TERPRI)
(PRIN1 "CONDITIONS:")
(TERPRI)
[PROG [(C (GETPROP PRO (QUOTE CONDITIONS)
  LOOP
    (COND
      ((NULL C)
        (RETURN))
      (T (SELECTQ (CAAR C)
        (*UNLESS* (PRETTYASSR (CADAR C)
          NIL 0.0))
        (*NOT* (PRETTYASSR (CADAR C)
          NIL -1.0))
        [*OR* (PRINT (QUOTE *OR*))
          (MAPC (CDAR C)
            (FUNCTION (LAMBDA (X)
              (TAB 3)
              (PRETTYASSR X)
              (PRETTYASSR (CAR C)))
            (SETQ C (CDR C))
            (GO LOOP)]
        (TERPRI)
```


SLOWER-THAN-A-MERCHANT MATCH-PLAT
OUTSIDE-ALL-LANES))

(PUTPROPS INHERIT CONDITIONS ((ALIAS *PLAT *UNKNOWN)
(TYPE *PLAT *TYP)
(ID *PLAT *ID1)
(ID-AMPLIFY *PLAT *IDMP)
(CLASS *PLAT *CLS)
(MEDIUM *PLAT *MED)))

(PUTPROPS NOT-LAST-SIGHTING CONDITIONS ((SIGHTING *PLAT *S1)
(SIGHTING *PLAT *S2)
(*NOT* (SAME-AS *S1 *S2))
(TOS *S1 *T1)
(TOS *S2 *T2)
(LESS-THAN *T1 *T2)
(*UNLESS* (NOT-LAST *S1))))

(PUTPROPS NOT-FIRST-SIGHTING CONDITIONS ((SIGHTING *PLAT *S1)
(SIGHTING *PLAT *S2)
(*NOT* (SAME-AS *S1 *S2))
(TOS *S1 *T1)
(TOS *S2 *T2)
(LESS-THAN *T2 *T1)
(*UNLESS* (NOT-FIRST *S1))))

(PUTPROPS FIRST-VIEW CONDITIONS ((SIGHTING *PLAT *S1)
(*UNLESS* (NOT-FIRST *S1))))

(PUTPROPS NOT-KNOWN-COMBATANT CONDITIONS ((CONTACT *CONT)
(SIGHTING *CONT *S1)
(*UNLESS* (WITHIN-REACH
*S1 *S2))
(SIGHTING *PLAT *S2)
(ID-AMPLIFY *PLAT MIL-BATTLE)
(*UNLESS* (OWNSHIP *PLAT))))

(PUTPROPS REACHABLE CONDITIONS ((CONTACT *CONT)
(SIGHTING *CONT *S1)
(SIGHTING *PLAT *S2)
(*NOT* (SAME-AS *PLAT *CONT))
(*UNLESS* (OWNSHIP *PLAT))
(SIMPLY-WITHIN-REACH *S1 *S2)
(*UNLESS* (BLOCKED-FROM *S1 *S2))))

(PUTPROPS SIMPLY-REACHABLE CONDITIONS ((CONTACT *CONT)
(SIGHTING *CONT *S1)
(SIGHTING *PLAT *S2)
(ID-AMPLIFY *PLAT MIL-BATTLE)
(*NOT* (SAME-AS *CONT *PLAT))
(*UNLESS* (OWNSHIP *PLAT))
(POSITION *S1 *P1)
(POSITION *S2 *P2)
(TOS *S1 *T1)
(TOS *S2 *T2)
(SWR *P1 *T1 *P2 *T2)))

```
(PUTPROPS POSS-RPT CONDITIONS ((PATROL *PTL)
                                (CONTACT *CONT)
                                (SIGHTING *CONT *S1)
                                (SIGHTING *PLAT *S2)
                                (ID-AMPLIFY *PLAT MIL-BATTLE)
                                (*UNLESS* (OWNSHIP *PLAT))
                                (SOURCE *S2 *PTL)
                                (*NOT* (SAME-AS *S1 *S2))
                                (*UNLESS* (DISSIMILAR *CONT *PLAT))))

(PUTPROPS BLOCKER CONDITIONS ((CONTACT *CONT)
                               (SIGHTING *CONT *S1)
                               (SIGHTING *PLAT *S2)
                               (ID-AMPLIFY *PLAT MIL-BATTLE)
                               (*NOT* (SAME-AS *CONT *PLAT))
                               (*UNLESS* (OWNSHIP *PLAT))
                               (PATROL *PTL)
                               (*UNLESS* (POSSIBLE-REPORT *CONT *PTL))
                               (SIGHTING *PTL *S3)
                               (NOT-LAST *S3)
                               (SUCCESSOR *S3 *S4)
                               (POSITION *S1 *P1)
                               (POSITION *S2 *P2)
                               (POSITION *S3 *P3)
                               (POSITION *S4 *P4)
                               (TOS *S1 *T1)
                               (TOS *S2 *T2)
                               (TOS *S3 *T3)
                               (TOS *S4 *T4)
                               (*OR* (CROSSPATHS *P1 *P2 *P3 *P4)
                                      (GRAZE *P1 *P2 *P3 *P4))
                               (*NOT* (WENT-BEFORE *P1 *T1 *P2 *T2 *P3 *T3 *P4 *T4))
                               (*NOT* (WENT-AFTER *P1 *T1 *P2 *T2 *P3 *T3 *P4 *T4))))

(PUTPROPS CREATEDETECT CONDITIONS ((SIGHTING *PLAT *SGT)
                                    (SOURCE *SGT EW)
                                    (*UNLESS* (DETECTION *PLAT))))

(PUTPROPS CREATECONTACT CONDITIONS ((SIGHTING *PLAT *SGT)
                                    (SOURCE *SGT RADAR)
                                    (*UNLESS* (CONTACT *PLAT))))

(PUTPROPS CREATEPLAT CONDITIONS ((SIGHTING *PLAT *SGT)
                                  (*UNLESS* (OWNSHIP *PLAT))
                                  (*UNLESS* (PLATFORM *PLAT))))

(PUTPROPS SMALL-CRAFT9 CONDITIONS ((CONTACT *WHO)
                                   (FIRST-SIGHTING *WHO *S1)
                                   (SOURCE *S1 RADAR)
                                   (RANGE *S1 *R1)
                                   (LESS-THAN *R1 8)
                                   (STRENGTH *S1 STRONG)))

(PUTPROPS SMALL-CRAFT6 CONDITIONS ((CONTACT *X)
                                   (SIGHTING *X *SIGHT)
                                   (NOT-FIRST *SIGHT)
                                   (RANGE *SIGHT *R))
```

(LESS-THAN *R 16)
(GREATER-THAN *R 9)
(STRENGTH *SIGHT WEAK)
(SPEED *SIGHT *SPD)
(*UNLESS* (GREATER-THAN *SPD 20)))

(PUTPROPS SMALL-CRAFT5 CONDITIONS ((CONTACT *WHO)
(SIGHTING *WHO *S1)
(NOT-FIRST *S1)
(SOURCE *S1 RADAR)
(RANGE *S1 *RANGE)
(LESS-THAN *RANGE 16)
(GREATER-THAN *RANGE 9)
(STRENGTH *S1 WEAK)
(SPEED *S1 *SPEED)
(GREATER-THAN *SPEED 20)))

(PUTPROPS SMALL-CRAFT4 CONDITIONS ((CONTACT *UNKNOWN)
(SIGHTING *UNKNOWN *SIGHTING1)
(LAND-DIST *SIGHTING1 *DIST)
(SOURCE *SIGHTING1 RADAR)
(RANGE *SIGHTING1 *RANGE)
(LESS-THAN *RANGE 9)
(GREATER-THAN *RANGE 3)
(STRENGTH *SIGHTING1 WEAK)
(LESS-THAN *DIST 50)))

(PUTPROPS SMALL-CRAFT3 CONDITIONS ((CONTACT *UNKNOWN)
(SIGHTING *UNKNOWN *SIGHTING)
(LAND-DIST *SIGHTING *DIST)
(SOURCE *SIGHTING RADAR)
(RANGE *SIGHTING *RANGE)
(LESS-THAN *RANGE 9)
(GREATER-THAN *RANGE 3)
(STRENGTH *SIGHTING WEAK)
(GREATER-THAN *DIST 50)))

(PUTPROPS SMALL-CRAFT2 CONDITIONS ((CONTACT *UNKNOWN)
(SIGHTING *UNKNOWN *SIGHTING)
(NOT-FIRST *SIGHTING)
(SOURCE *SIGHTING RADAR)
(STRENGTH *SIGHTING WEAK)
(SPEED *SIGHTING *SPEED)
(*UNLESS* (GREATER-THAN *SPEED 3))))

(PUTPROPS SMALL-CRAFT1 CONDITIONS ((CONTACT *UNKNOWN)
(SIGHTING *UNKNOWN *SIGHTING)
(NOT-FIRST *SIGHTING)
(SOURCE *SIGHTING RADAR)
(RANGE *SIGHTING *RANGE)
(LESS-THAN *RANGE 3)
(STRENGTH *SIGHTING WEAK)
(SPEED *SIGHTING *SPEED)
(GREATER-THAN *SPEED 3)))

(PUTPROPS ID-LANE CONDITIONS ((SIGHTING *SHIP *SIGHTING)
(MERCHANTLANE *LANE)

(PLATFORM *SHIP)
(LOCATION *LANE *LANELOC)
(POSITION *SIGHTING *POS)
(IN-LANE *LANELOC *POS)))

(PUTPROPS INSIDE-A-STORM CONDITIONS ((SIGHTING *SHIP *SIGHTING)
(PLATFORM *SHIP)
(STORM *STORM)
(POSITION *SIGHTING *POS)
(LOCATION *STORM *STMLOC)
(INSIDE *POS *STMLOC)))

(PUTPROPS CLOSE-POPUP CONDITIONS ((CONTACT *SHIP)
(FIRST-SIGHTING *SHIP *SIGHTING)
(RANGE *SIGHTING *RANGE)
(LESS-THAN *RANGE 12)))

(PUTPROPS DISTANT-POPUP CONDITIONS ((CONTACT *SHIP)
(FIRST-SIGHTING *SHIP *SIGHTING)
(RANGE *SIGHTING *RANGE)
(GREATER-THAN *RANGE 30)))

(PUTPROPS COURSE-CHANGED CONDITIONS ((CONTACT *SHIP)
(SIGHTING *SHIP *SIGHTING1)
(NOT-FIRST *SIGHTING1)
(NOT-LAST *SIGHTING1)
(SUCCESSOR *SIGHTING1 *SIGHTING2)
(COURSE *SIGHTING1 *COURSE1)
(COURSE *SIGHTING2 *COURSE2)
(*UNLESS* (
ROUGHLY-THE-SAME-COURSE-AS *COURSE1 *COURSE2))))

(PUTPROPS SPEED-CHANGED CONDITIONS ((CONTACT *SHIP)
(SIGHTING *SHIP *SIGHTING)
(NOT-FIRST *SIGHTING)
(NOT-LAST *SIGHTING)
(SUCCESSOR *SIGHTING *SIGHTING2)
(SPEED *SIGHTING *SPEED1)
(SPEED *SIGHTING2 *SPEED2)
(*UNLESS* (ROUGHLY-THE-SAME-SPEED-AS
*SPEED1 *SPEED2))))

(PUTPROPS FASTER-THAN-A-MERCHANT CONDITIONS ((CONTACT *SHIP)
(SIGHTING *SHIP *SIGHTING)
(NOT-FIRST *SIGHTING)
(SPEED *SIGHTING *SPEED)
(GREATER-THAN *SPEED 25)))

(PUTPROPS SLOWER-THAN-A-MERCHANT CONDITIONS ((CONTACT *SHIP)
(SIGHTING *SHIP *SIGHTING)
(NOT-FIRST *SIGHTING)
(SPEED *SIGHTING *SPEED)
(LESS-THAN *SPEED 9)))

(PUTPROPS MATCH-PLAT CONDITIONS ((SIGHTING *PLAT1 *SGT1)
(NOT-FIRST *SGT1)
(SIGHTING *PLAT2 *SGT2))

```
( *NOT* (SAME-AS *PLAT1 *PLAT2))
(*UNLESS* (NOT-LAST *SGT2))
(COURSE *SGT1 *CRS1)
(SPEED *SGT1 *SPD1)
(POSITION *SGT1 *POS1)
(TOS *SGT1 *T1)
(POSITION *SGT2 *POS2)
(TOS *SGT2 *T2)
(LESS-THAN *T2 *T1)
(COURSEFROM *POS2 *POS1 *CRS2)
(SPEEDFROM *POS2 *T2 *POS1 *T1 *SPD2)
(ROUGHLY-THE-SAM-COURSE-AS *CRS1 *CRS2)
(ROUGHLY-THE-SAME-SPEED-AS *SPD1 *SPD2)
))
```

[illegible]

```
(PUTPROPS INHERIT ACTIONS ((TYPE *UNKNOWN *TYP)
                             (ID *UNKNOWN *ID1)
                             (ID-AMPLIFY *UNKNOWN *IDMP)
                             (CLASS *UNKNOWN *CLS)
                             (MEDIUM *UNKNOWN *MED)))
```

```
(PUTPROPS NOT-LAST-SIGHTING ACTIONS ((NOT-LAST *S1)))
```

(PUTPROPS NOT-FIRST-SIGHTING ACTIONS ((NOT-FIRST *S1)))

```
(PUTPROPS FIRST-VIEW ACTIONS ((FIRST-SIGHTING *PLAT *S1)))
```

(PUTPROPS NOT-KNOWN-COMBATANT ACTIONS ((TYPE *CONT MERCHANT)))

```
(PUTPROPS REACHABLE ACTIONS ((WITHIN-REACH *S1 *S2)))
```

```
(PUTPROPS SIMPLY-REACHABLE ACTIONS ((SIMPLY-WITHIN-REACH *S1 *S2)))
```

(PUTPROPS POSS-RPT ACTIONS ((POSSIBLE-REPORT *CONT *PTL)))

```
(PUTPROPS BLOCKER ACTIONS ((BLOCKED-FROM *S1 *S2)))
```

```
(PUTPROPS CREATEDETECT ACTIONS ((DETECTION *PLAT)))
```

```
(PUTPROPS CREATECONTACT ACTIONS ((CONTACT *PLAT)))
```

```
(PUTPROPS CREATEPLAT ACTIONS ((PLATFORM *PLAT)))
```

```
(PUTPROPS SMALL-CRAFT9 ACTIONS ((TYPE *WHO SUB)
                                (MODE *WHO SURFACE)))
```

[illegible]

(PUTPROPS SMALL-CRAFT5 ACTIONS ((*OR* (TYPE *WHO SUB)
(TYPE *WHO PATROL))))

(PUTPROPS SMALL-CRAFT4 ACTIONS ((*OR* (TYPE *UNKNOWN SUB)
(TYPE *UNKNOWN SHORE-PATROL)
(TYPE *UNKNOWN PLEASURE)
(TYPE *UNKNOWN COMMERCIAL)
(TYPE *UNKNOWN LANDING))))

(PUTPROPS SMALL-CRAFT3 ACTIONS ((TYPE *UNKNOWN SUB)))

(PUTPROPS SMALL-CRAFT2 ACTIONS ((*OR* (TYPE *UNKNOWN DEBRIS)
(TYPE *UNKNOWN SUB)
(TYPE *UNKNOWN BUOY))))

(PUTPROPS SMALL-CRAFT1 ACTIONS ((TYPE *UNKNOWN SUB)
(*OR* (MODE *UNKNOWN PERISCOPE)
(MODE *UNKNOWN SNORKEL))))

(PUTPROPS ID-LANE ACTIONS ((INSIDE-A-MERCHANTLANE *SIGHTING)
(*REPORT* *SHIP
" was sighted in the merchant lane "
*LANE)))

(PUTPROPS INSIDE-A-STORM ACTIONS ((TYPE *SHIP MERCHANT)
(*REPORT* *SHIP
" was sighted inside "
*STORM)))

(PUTPROPS CLOSE-POPUP ACTIONS ((TYPE *SHIP MERCHANT)))

(PUTPROPS DISTANT-POPUP ACTIONS ((TYPE *SHIP MERCHANT)))

(PUTPROPS COURSE-CHANGED ACTIONS ((TYPE *SHIP MERCHANT)))

(PUTPROPS SPEED-CHANGED ACTIONS ((TYPE *SHIP MERCHANT)))

(PUTPROPS FASTER-THAN-A-MERCHANT ACTIONS ((TYPE *SHIP MERCHANT)))

(PUTPROPS SLOWER-THAN-A-MERCHANT ACTIONS ((TYPE *SHIP MERCHANT)))

(PUTPROPS MATCH-PLAT ACTIONS ((ALIAS *PLAT2 *PLAT1)))

(PUTPROPS OUTSIDE-ALL-LANES ACTIONS ((TYPE *SHIP MERCHANT)))

(PUTPROPS INHERIT CONF 1.0)

(PUTPROPS NOT-LAST-SIGHTING CONF 1.0)

(PUTPROPS NOT-FIRST-SIGHTING CONF 1.0)

(PUTPROPS FIRST-VIEW CONF .99)

(PUTPROPS NOT-KNOWN-COMBATANT CONF .45)

(PUTPROPS REACHABLE CONF .97)

(PUTPROPS SIMPLY-REACHABLE CONF .98)

(PUTPROPS POSS-RPT CONF .95)

(PUTPROPS BLOCKER CONF .9)

(PUTPROPS CREATEDETECT CONF 1.0)

(PUTPROPS CREATECONTACT CONF 1.0)

(PUTPROPS CREATEPLAT CONF 1.0)

(PUTPROPS SMALL-CRAFT9 CONF .5)

(PUTPROPS SMALL-CRAFT6 CONF .15)

(PUTPROPS SMALL-CRAFT5 CONF .3)

(PUTPROPS SMALL-CRAFT4 CONF .1)

(PUTPROPS SMALL-CRAFT3 CONF .35)

(PUTPROPS SMALL-CRAFT2 CONF .12)

(PUTPROPS SMALL-CRAFT1 CONF .6)

(PUTPROPS ID-LANE CONF 1.0)

(PUTPROPS INSIDE-A-STORM CONF -.25)

(PUTPROPS CLOSE-POPUP CONF -.2)

(PUTPROPS DISTANT-POPUP CONF -.2)

(PUTPROPS COURSE-CHANGED CONF -.3)

(PUTPROPS SPEED-CHANGED CONF -.3)

(PUTPROPS FASTER-THAN-A-MERCHANT CONF -.25)

(PUTPROPS SLOWER-THAN-A-MERCHANT CONF -.15)

(PUTPROPS MATCH-PLAT CONF .5)

(PUTPROPS OUTSIDE-ALL-LANES CONF -.08)

(DECLARE: DONTCOPY

(FILEMAP (NIL (936 4519 (DEFINEPD 948 . 1975) (MAKEPD 1979 . 2509) (

FANCYPROD 2513 . 4150) (LINEREAD 4154 . 4516))))))

STOP

(FILECREATED " 6-Aug-79 20:15:20" <PMORRIS>STREAM.LSP.37 7631

changes to: UNFREEZE

previous date: " 6-Aug-79 14:19:01" <PMORRIS>STREAM.LSP.36)

(PRETTYCOMPRINT STREAMCOMS)

```
(RPAQQ STREAMCOMS ((FNS * STREAMFNS)
  (VARS (MAPRETALIST NIL)
    (FREEZEFLG NIL)
    (FREEZELST NIL))))
```

```
(RPAQQ STREAMFNS (ENDSTREAM FREEZE MAPSTREAM NEWSTREAM MAPRETRIEVE
  MAPRETDO RETPULSED0 RETRIEVES PREPALIST
  RETSTREAM GETMRVAL SOMEPULSE STRIPSTREAM
  PULSAR PULSE PUTSTREAM UNFREEZE))
```

(DEFINEQ

[240]

```
(ENDSTREAM
  [LAMBDA (S)
```

```
(* edited:
  " 6-Aug-79 14:08")
```

```
(* This has an effect like putting an end marker on
  a stream. It actually discards the suspensions and
  replaces them by the marker T, which informs
  MAPSTREAM not to place new suspensions on the
  stream.)
```

(RPLACD S T)

[241]

```
(FREEZE
  [LAMBDA NIL
```

```
(* edited:
  " 3-Aug-79 17:32")
```

```
(SETQ FREEZEFLG T)
(QUOTE Brr..)
```

[242]

```
(MAPSTREAM
  [LAMBDA (MAPSTREAMX MAPSTREAMINFO MAPSTREAMFN)
```

```
(* edited:
  " 6-Aug-79 14:13")
```

```
(PROG NIL
  [MAPC (CAAR MAPSTREAMX)
    (FUNCTION (LAMBDA (X)
      (APPLY* MAPSTREAMFN X MAPSTREAMINFO))
    (COND
      ((NEQ (CDR MAPSTREAMX)
        T)
```

(TCONC (CDR MAPSTREAMX)
(CONS MAPSTREAMINFO MAPSTREAMFN))

[243]

(NEWSTREAM
[LAMBDA NIL

(* edited:
"11-Apr-79 17:09")

(CONS (CONS)
(CONS))

[244]

(MAPRETRIEVE
[LAMBDA (MAPRETX MAPRETINFO MAPRETFN)

(* edited:
" 6-Jul-79 16:18")

(MAPSTREAM (RETSTREAM MAPRETX)
(CONS (CONS MAPRETALIST MAPRETINFO)
(CONS MAPRETX MAPRETFN))
(FUNCTION MAPRETD0))

[245]

(MAPRETD0
[LAMBDA (SELT AI)

(* edited:
" 6-Jul-79 16:23")

(SOMEPULSE (GETPULSAR SELT)
(CONS SELT AI)
(FUNCTION RETPULSED0))

[246]

(RETPULSED0
[LAMBDA (SELT AI)

(* edited:
" 6-Jul-79 16:43")

(PROG ((SELT (CAR SELT AI))
(AI (CDR SELT AI))
ASS MAPRETALIST MAPRETINFO MAPRETX MAPRETFN)
(DECLARE (SPECVARS MAPRETALIST))
(SETQ ASS (GETUPLE SELT))
(SETQ MAPRETALIST (CAAR AI))
(SETQ MAPRETINFO (CDAR AI))
(SETQ MAPRETX (CADR AI))
(SETQ MAPRETFN (CDDR AI))
(SETQ MAPRETALIST (PREPALIST MAPRETX ASS MAPRETALIST))
(RETURN (APPLY* MAPRETFN SELT MAPRETINFO)))

[247]

(RETRIEVES
[LAMBDA (AT OBJ VAL SEL)

(* edited:
"27-Jul-79 15:25")

(PROG ((SPEC (QUOTE (NIL NIL NIL)))
ASS ELT ANS LAST ONEFLG ASSES)
(RPLACA SPEC AT)
(RPLACA (CDR SPEC)
OBJ)

```

[COND
  (VAL (RPLACA (CDDR SPEC)
              VAL))
  (T (SETQ LAST (LAST SPEC))
      (RPLACD (CDR SPEC)
              VAL))
[OR SEL (SETQ SEL (COND
  ((EQ AT (QUOTE *))
   1)
  ((EQ OBJ (QUOTE *))
   2)
  ((EQ VAL (QUOTE *))
   3)
  (T 2]
(SETQ ASSES (STRIPSTREAM (GETSH SPEC)))
[COND
  [(ILESSP (for X in SPEC count (EQ X (QUOTE *)))
   2)
   (SETQ ANS (for ASS in ASSES
               collect (CAR (NTH (GETUPLE ASS)
                                SEL)
                           (T (for ASS in ASSES do (SETQ ELT
                                                         (CAR (NTH (GETUPLE ASS)
                                                                SEL))))
                           (COND
                             ((NOT (MEMBER ELT ANS))
                              (SETQ ANS (CONS ELT ANS)
                                     (OR VAL (NCONC SPEC LAST))
                                     (RETURN ANS))

```

[248]

```

(PREPALIST
 [LAMBDA (CON ASS ALIST)
   (* edited:
    " 6-Jul-79 17:55")
   (for C in CON as A in ASS do [COND
     ((AND (VAR? C)
            (NOT (ASSOC C ALIST)))
      (SETQ ALIST
              (CONS (CONS C A)
                    ALIST)
      finally (RETURN ALIST))

```

[249]

```

(RETSTREAM
 [LAMBDA (C)
   (* edited:
    "29-Jun-79 15:44")

```

(* This function returns the stream corresponding to a rule condition. It constructs a specification from the condition, taking into account the variable bindings. It reuses a scratchlist (of 10 elements) for efficiency, chopping off the piece that it doesn't need. After the stream is obtained, the scratchlist is restored to full size.)

```

(PROG ((SCRATCH (QUOTE (0 0 0 0 0 0 0 0 0 0)))
      PTR FOLLOW S XASSOC)
      (SETQ PTR SCRATCH)
      [MAPC C (FUNCTION (LAMBDA (X)
                        (RPLACA PTR (COND
                                [(VAR? X)
                                 (SETQ XASSOC (ASSOC X MAPRETALIST))
                                 (COND
                                  (XASSOC (CDR XASSOC))
                                  (T (QUOTE *])
                                (T X)))
                        (SETQ FOLLOW PTR)
                        (SETQ PTR (CDR PTR)
                        (RPLACD FOLLOW NIL)
                        (SETQ S (GETSH SCRATCH))
                        (RPLACD FOLLOW PTR)
                        (RETURN S)])

```

[250]

```

(GETMRVAL
 [LAMBDA (X COPYFLG)
 (* edited:
 "25-Jul-79 13:49")
 (SUBLIS MAPRETALIST X COPYFLG)])

```

[251]

```

(SOMEPUSE
 [LAMBDA (PULSAR PULSARDATA SOMEPUSEFN)
 (* edited:
 " 5-Jul-79 19:06")
 (OR (APPLY* SOMEPUSEFN PULSARDATA)
      (TCONC PULSAR (CONS SOMEPUSEFN PULSARDATA))

```

[252]

```

(STRIPSTREAM
 [LAMBDA (S)
 (* edited:
 "29-Jun-79 17:52")
 (CAAR S)])

```

[253]

```

(PULSAR
 [LAMBDA NIL
 (* edited:
 " 5-Jul-79 18:49")
 (CONS)])

```

[254]

```

(PULSE
 [LAMBDA (PULSAR)
 (* edited:
 " 6-Jul-79 13:17")
 (PROG ((CELL (CONSTANT (CONS)))
      PTR)
      (SETQ PTR (RPLACD CELL (CAR PULSAR)))
      LOOP(COND
            [(NULL (CDR PTR))

```

```

(RPLACA PULSAR (CDR CELL))
(RPLACD PULSAR (COND
  ((CAR PTR)
   PTR]
(T [COND
  ((APPLY* (CAADR PTR)
            (CDADR PTR))
   (RPLACD PTR (CDDR PTR)))
  (T (SETQ PTR (CDR PTR)
    (GO LOOP]))

```

[255]

```

(PUTSTREAM
  [LAMBDA (S X)

```

```

(* edited:
  " 6-Aug-79 14:18")

```

```

(COND
  ((EQ (CDR S)
        T)
   (HELP "Can't put into ended stream - PUTSTREAM")))
(TCONC (CAR S)
  X)
(MAPC (CADR S)
  (FUNCTION (LAMBDA (SUSP)
    (COND
      (FREEZEFLG (SETQ FREEZELST (CONS (CONS X SUSP)
        FREEZELST)))
      (T (APPLY* (CDR SUSP)
                  X
                  (CAR SUSP)]

```

[256]

```

(UNFREEZE
  [LAMBDA NIL

```

```

(* edited:
  " 6-Aug-79 20:15")

```

```

  (SETQ FREEZEFLG NIL)
  [MAPC (DREVERSE FREEZELST)
    (FUNCTION (LAMBDA (XSUSP)
      (APPLY* (CDDR XSUSP)
              (CAR XSUSP)
              (CADR XSUSP]
    (SETQ FREEZELST NIL)
    (QUOTE Ahh..)]
)

```

```

(RPAQ MAPRETALIST NIL)

```

```

(RPAQ FREEZEFLG NIL)

```

```

(RPAQ FREEZELST NIL)

```

```

(DECLARE: DONTCOPY

```

```

  (FILEMAP (NIL (526 7533 (ENDSTREAM 538 . 894) (FREEZE 898 . 1036) (
MAPSTREAM 1040 . 1535) (NEWSTREAM 1539 . 1666) (MAPRETRIEVE 1670 . 1913)
  (MAPRETDO 1917 . 2128) (RETPULSED0 2132 . 2803) (RETRIEVES 2807 . 4155)
  (PREPALIST 4159 . 4526) (RETSTREAM 4530 . 5535) (GETMRVAL 5539 . 5681)
  (SOMEPUULSE 5685 . 5870) (STRIPSTREAM 5874 . 5988) (PULSAR 5992 . 6099) (

```

<PMORRIS>STREAM.LSP.37

Page 169

PULSE 6103 . 6750) (PUTSTREAM 6754 . 7211) (UNFREEZE 7215 . 7530))))
STOP

(FILECREATED "21-Aug-79 11:08:04" <RBECHTAL>TOPLEVEL..13 6972

changes to: WELCOME

previous date: " 9-Aug-79 13:23:33" <RBECHTAL>TOPLEVEL..12)

(PRETTYCOMPRINT TOPLEVELCOMS)

```
(RPAQQ TOPLEVELCOMS ((VARS * TOPLEVELVARS)
                      (FNS * TOPLEVELFNS)
                      (P (MINFS 512 2))))
```

(RPAQQ TOPLEVELVARS (RESULTLIST DUALFLG))

(RPAQQ RESULTLIST NIL)

(RPAQQ DUALFLG NIL)

```
(RPAQQ TOPLEVELFNS (ADDIS CKCONFIGURATION EXLOOP INCLUDEPLAT PARTING
                          STAMMER STARTUP STUFFLN WAITER WELCOME))
```

(DEFINEQ

[257]

```
(ADDIS
 [LAMBDA (SN)
```

```
(* edited:
 " 6-Aug-79 13:44"
 (* ATTIS places a single
 sighting of a platform
 into the display file.)
```

```
(PROG ((PLT (GETATTB (QUOTE SIGHTING)
                     SN))
       (POS (GETATT (QUOTE POSITION)
                     SN))
       (TIM (GETATT (QUOTE TOS)
                     SN)))
 (DISPLAY PLT (CAAR POS)
           (CADAR POS)
           TIM))
```

[258]

```
(CKCONFIGURATION
 [LAMBDA NIL
```

(PROG NIL

```
(* edited:
 "31-Jul-79 16:23"
 (* CKCONFIGURATION
 determines the terminal
 configuration and
 initializes the display
 routines.)
```

```
(COND
 ((TEKTEST)
  (DSPGRAB)
  (DSPINIT)
  (PRIN1 "Do you want a map? ")
```

```

(COND
  ((EQP (CHCON1 (ASKUSER))
    89)
    (DSPMAP)))
(CLEARBUF)
(SETQ DISPLAYFLG T)
(STARTUP))
(T (PRIN1 "Do you have a Tektronix available for display? ")
  (COND
    ((EQP (CHCON1 (ASKUSER))
      89)
      (CLEARBUF)
      (SETQ DUALFLG T)
      (PRIN1 "What is the Tek terminal number? ")
      (OR (DSPGRAB (READ))
        (HELP "Failed to initialize display terminal: "
          (DSPTTYSTR)))
      (CLEARBUF)
      (DSPINIT)
      (PRIN1 "Do you want a map? ")
      (COND
        ((EQP (CHCON1 (ASKUSER))
          89)
          (DSPMAP)))
      (CLEARBUF)
      (SETQ DISPLAYFLG T)
      (STARTUP))
    (T (CLEARBUF))
  )

```

[259]

```

(EXLOOP
  [LAMBDA NIL

```

```

(* edited:
  " 6-Aug-79 13:22")

```

(* EXLOOP is where all the real work gets done. MSGMTR reads messages and places them into memory, and returns an indication of what should be done next. Unless MSGMTR returns NIL (out of messages) or IGNORE (uninteresting), the results of any rule firings are printed, and the explanation system is called. Notice that with the stream oriented rule interpreter, there is no distinct "rule interpretation" cycle or function call.)

```

(PROG (MSGFLG)
  EXLP(SETQ MSGFLG (MSGMTR))
  (COND
    ((EQ MSGFLG (QUOTE IGNORE))
      (GO EXLP))
    (MSGFLG (RESOUT)
      (EXPLAIN)
      (GO EXLP))
    (T (RETURN))
  )

```

[260]

(INCLUDEPLAT
[LAMBDA (PNE)

(* edited:
" 6-Aug-79 13:24")

(* This places any previously existing platform
sightings into the display file.
Used for initialization for "snapshot" memories.)

(MAPC (RETRIEVES (QUOTE SIGHTING)
PNE
(QUOTE *)
3)
(FUNCTION ADDIS])

[261]

(PARTING
[LAMBDA NIL

(* edited:
" 6-Aug-79 13:25")

(* PARTING cleans up after a STAMMER run.
Kills the display job (if any), and notifies the
user of the end of run.)

(PRIN1 " Thank you for your interest in the STAMMER system."
(TERPRI)
(COND
(DSPLAYFLG (FKKILL)
(DSPRELD])

[262]

(STAMMER
[LAMBDA NIL

(* edited:
" 6-Aug-79 13:27")

(* This is it! Start rule interpretation by doing an
APPLYRULE to all the rules, greet the user, do
EXLOOP, and leave. Simplicity itself.)

(MAPC PRODUCTIONS (FUNCTION APPLYRULE))
(WELCOME)
(EXLOOP)
(PARTING))

[263]

STARTUP
[LAMBDA NIL

(* edited:
" 6-Aug-79 13:35")

STARTUP calls functions to place existing

platforms and merchantlanes into the display file,
as appropriate. Particularly oriented for
intermediate memory saves.)

```
(MAPC (RETRIEVES (QUOTE PLATFORM)
  (QUOTE *)
  NIL 2)
  (FUNCTION INCLUDEPLAT))
(MAPC (RETRIEVES (QUOTE MERCHANTLANE)
  (QUOTE *)
  NIL 2)
  (FUNCTION STUFFLN))
```

[264]

```
(STUFFLN
  [LAMBDA (MLN)
```

```
(* edited:
  " 6-Aug-79 13:36")
```

(* STUFFLN places all of the locations of a merchant
lane into the display file. Like INCLUDEPLAT.)

```
(DSPADDTRH MLN (QUOTE ML)
  (QUOTE XX))
(MAPC (GETATT (QUOTE LOCATION)
  MLN)
  (FUNCTION (LAMBDA (VER)
    (DSPADDINC MLN (CAR VER)
      (CADR VER)
      0.01))
```

[265]

```
(WAITER
  [LAMBDA NIL
```

```
(* NOBIND
  "22-Dec-78 16:33")
```

(* WAITER is used to introduce a user-controllable
delay in single terminal display mode.)

```
(COND
  ((NOT DUALFLG)
    (TERPRI)
    (TERPRI)
    (TERPRI)
    (TERPRI)
    (TEKWAIT)
    (ASKUSER 5 (QUOTE %
  )
    "<CR> to continue, <SPACE> to wait:"
    (QUOTE ( "
  "]))
```

[266]

(WELCOME
[LAMBDA NIL

(* edited:
"21-Aug-79 11:07")

(* WELCOME is the "first" thing that gets done when running STAMMER. (Actually, the rules are initialized first.) It's a good place to put any initialization stuff. Now, it is used to allow the user to select a message file.)

```
(PROG (NEWFL)
  (PRIN1 " Welcome to version 2 of the STAMMER TSA system.")
  (TERPRI)
  (PRIN1
    "What file would you like to take messages from?
(Default is ")
    (PRIN1 MSGFILE)
    (PRIN1 "): ")
    (SETQ NEWFL (LINEREAD))
    [COND
      ((NULL NEWFL))
      (T (SETQ MSGFILE (CAR NEWFL)
    (TERPRI)
    [MAPC ASSERTIONS (FUNCTION (LAMBDA (TB)
      (PUTPROP TB (QUOTE TDB)
        T]
    (CKCONFIGURATION])
  )
(MINFS 512 2)
(DECLARE: DONTCOPY
  (FILEMAP (NIL (516 6933 (ADDIS 528 . 1040) (CKCONFIGURATION 1044 . 2141)
    (EXLOOP 2145 . 3082) (INCLUDEPLAT 3086 . 3527) (PARTING 3531 . 3987) (
    STAMMER 3991 . 4423) (STARTUP 4427 . 5015) (STUFFLN 5019 . 5493) (WAITER
    5497 . 5877) (WELCOME 5881 . 6930))))))
STOP
```

1.	STAMMER	WELCOME	CKCONFIGURATION	TEKTEST	PRINCHAR
2.				DSPINIT	FKINIT FKRCAS
3.					FKJSYS ASSEMBLE
4.					FKJSYSARG
5.					AC
6.					FKTTYSET FKJSYS {3}
7.					FKSW FKJSYS {3}
8.					FKHALT
9.				DSPQUIET	FKCALL FKARRAYP FKSHR
10.					NOFORK FKINIT {2}
11.					FKWAIT FKJSYS {3}
12.					FKIDPB
13.					FKCALLERR
14.					PUTTYP
15.					FKSACS
16.					FKSW {7}
17.					FKACSRETURN
18.					FKRCAS
19.					FKHNDL
20.					FKHT
21.					FKACS
22.					FKSYM FKSACS
23.					FKSW {7}
24.					FKRCAS
25.					FKHT
26.					NOFORK {10}
27.					FKSYMACS
28.					FKHNDL
29.					GETRADIX50
30.					FKSYMPUT {a}
31.					FKCATYPE
32.					FKSR FKJSYS {3}
33.					FKARRAYTYPE
34.					FKSHR
35.					FKRTN ASSEMBLE
36.					FKJSYS {3}
37.				BKDSPBUF	FKJSYS {3}
38.				DSPCNVRT	CRUNCH
39.				FKSETVAL	NOFORK {10}
40.					FKWAIT {11}
41.					FKSACS
42.					FKSW {7}
43.					FKACSRETURN
44.					FKHNDL
45.					FKHT

```

46. | | | | | FKACS
47. | | | | | FKSYS {22}
48. | | | | | FKIDPB
49. | | | | | FKCALL {9}
50. | | | | | DSPTTY FKCALL {9}
51. | | | | | FKJSYS {3}
52. | | | | | DSPTTYSTR DECSAMEDIGITS {b}
53. | | | | | DSPMAP FKSETVAL {39}
54. | | | | | STARTUP RETRIEVES STRIPSTREAM
55. | | | | | GETSH GETH LOCH {c}
56. | | | | | PUTH NEWHASH {d}
57. | | | | | LOCH {c}
58. | | | | | NEWSTREAM
59. | | | | | GETUPLE
60. | | | | | INCLUDEPLAT RETRIEVES {54}
61. | | | | | ADDIS DISPLAY {e}
62. | | | | | GETATTB {f}
63. | | | | | GETATT {g}
64. | | | | | STUFFLN DSPADDTRH FKCALL {9}
65. | | | | | DSPCNVRT {38}
66. | | | | | GETATT {g}
67. | | | | | DSPADDINC FKCALL {9}
68. | | | | | DSPCNVRT {38}
69. | | | | | DSPGRAB OCTSAMEDIGITS OCTSAMEDIGITS {69}
70. | | | | | FKJSYS {3}
71. | | | | | DSPTTYSTR {52}
72. | | | | | LINEREAD
73. | | | | | EXLOOP RESOUT RESULTPRINTER GETUPLE
74. | | | | | PRETTYASSR ASSRPRINT
75. | | | | | EXPLAIN QHTAKE QHFOLLOW QHCLEAR
76. | | | | | QHPREP QHPREP {76}
77. | | | | | QHMAKE QHPUT
78. | | | | | QHMAKE {77}
79. | | | | | QHGET
80. | | | | | QHMAKE {77}
81. | | | | | QHSHOW QHPREP {76}
82. | | | | | QHFOLLOW {75}
83. | | | | | QHASK BEEP
84. | | | | | QHLIST QHLIST {84}
85. | | | | | QHGET
86. | | | | | QHGET
87. | | | | | RETRIEVES {54}
88. | | | | | NEWVALOBJ GETUPLE
89. | | | | | MSGMTR BEYONDINTEREST
90. | | | | | INTERPOLABLE

```

91.		FREEZE		
92.		UNFREEZE	apply	
93.		OWNMSG	CASSERT	SAVEPULSAR PULSAR
94.				SERT ADDH PUTSH PUTSTREAM apply
95.				GETSH {55}
96.				ENDSTREAM
97.				MATCHER
98.				BUMP
99.				GETSH {55}
100.				GETSTRIP STRIPSTREAM
101.				GETH {55}
102.				
103.				DISPLAY {e}
104.				NEWSYM
105.				DESCRIBMSG WAITER TEKWAIT
106.				DSPCMD FKCALL {9}
107.				GRATEK TEKCOM
108.				PRINCHAR
109.				MONTEK TEKCOM
110.				CENTROID
111.				WEATHERMSG CASSERT {93}
112.				DSPADDTRH {64}
113.				DSPADDINC {67}
114.				SENSORMSG CASSERT {93}
115.				DISPLAY {e}
116.				NEWSYM
117.				EWMSG CASSERT {93}
118.				DISPLOB DISPLAY {e}
119.				MIDP TWO-PLACE
120.				NEWSYM
121.				GETPOINT FIXLONG
122.				OWNPOS PLATPOS RETRIEVER GETSTRIP {100}
123.				RETVAR VAR?
124.				GETUPLE
125.				VAR?
126.				GETATT {g}
127.				PREDICTPOS GETATT {g}
128.				ESTIMATE {h}
129.				ONEPOINT {i}
130.				APPLYRULE SWEEPER apply
131.				SWEEPER {129}
132.				ORHACK SWEEPER {129}
133.				NOTHACK ORACLEHACK VAR?
134.				apply
135.				CASSERT {93}
				GETMRVAL

36.			MAPRETRIEVE	MAPSTREAM	apply
37.				RETSTREAM	VAR?
38.					GETSH {55}
39.				MAPRETDO	SOMEPUSE apply
40.					GETPULSAR
41.					RETPULSEDO {j}
42.			GETCON	GETMB	GETMARK
43.				BMEAS	MARKON
44.				BLFN	GETMD GETMARK
45.					DMEAS {k}
46.					GETCON {142}
47.					GETMB {142}
48.				DLFN	GETMB {142}
49.					GETCON {142}
50.					GETMD {144}
51.				GETMD	{144}
52.				GETCON	{142}
53.				SWEEPER	{129}
54.			UNLESSHACK	ORACLEHACK	{132}
55.				CASSERT	{93}
56.				STRIPSTREAM	
57.				RETSTREAM	{137}
58.				MESSAGE1	VAR?
59.					GETMRVAL
60.				MAPRETRIEVE	{136}
61.				GETCON	{142}
62.				SWEEPER	{129}
63.			ANDHACK	ORACLEHACK	{132}
64.				MAPRETRIEVE	{136}
65.				GETCON	{142}
66.				SWEEPER	{129}
67.			CONSTRUCT	ORBUILD	CONSTRUCT {167}
68.				JUSTBUILD	SAVEPULSAR {93}
69.					SERT {94}
70.					PULSE apply
71.					MESSAGE1 {158}
72.					GETSTRIP {100}
73.					GETPULSAR
74.				MESSAGE1	{158}
75.				GETMRVAL	
76.	PARTING	FKKILL	FKJSYS	{3}	
77.			FKHNDL		
78.			FKPROG		
79.			FKJFN		
80.			FKDDT	NOFORK {10}	

181.			FKDDT_	
182.			FKHNDL	
183.			FKDDT {180}	
184.			FKJSYS {3}	
185.			FKSW {7}	
186.			FKTTYSET {6}	
187.		FKSHR		
188.		DSPRELD FKJSYS {3}		
-----				overflow - a
189.		FKSYMPUT FKHT_		
-----				overflow - b
190.		DECSAMEDIGITS DECSAMEDIGITS {190}		
-----				overflow - c
191.		LOCH PREHASH PREHASH {191}		
192.		NEXTH		
-----				overflow - d
193.		NEWHASH CREATH		
194.		MAPH apply		
195.		PUTH {56}		
-----				overflow - e
196.		DISPLAY DISPCHECK		
197.		DSPADDTRH {64}		
198.		DISPMARK		
199.		DSPADDINC {67}		
200.		MELD		
201.		IDENT RETRIEVER {121}		
202.		GREATESTPROB GETCON {142}		
203.		MEDIUM RETRIEVER {121}		
204.		GREATESTPROB {202}		
-----				overflow - f
205.		GETATTB GETUPLE		
206.		STRIPSTREAM		
207.		GETSH {55}		
-----				overflow - g
208.		GETATT GETUPLE		
209.		STRIPSTREAM		
210.		GETSH {55}		
-----				overflow - h
211.		ESTIMATE SPAN NEAREST DISTANCE SUBTEND		
212.		GETATT {208}		
213.		AUXINTERPOL FIXLONG		
-----				overflow - i
214.		ONEPOINT GETATT {208}		
215.		SPAN {211}		
216.		CENTROID		

217. AUXINTERPOL {213}

----- overflow - j

218.RETPULSEDO apply

219. GETUPLE

220. PREPALIST VAR?

----- overflow - k

221.DMEAS MARKON

222. BLFN {144}

223. DLFN {148}

@end(verbatim)

ADDH[ARGS,NEWVAL]
calls: PUTSH
called by: SERT

ADDIS[SN]
calls: DISPLAY,GETATTB,GETATT
called by: INCLUDEPLAT
binds: PLT,POS,TIM

ANDHACK[CONDITIONS,ACTIONS,EV]
calls: ORACLEHACK,MAPRETRIEVE,LIST,FUNCTION,GREATERP,GETCON,SWEEPER,
CONS
called by: SWEEPER
binds: P,X,CLIST

APPLYRULE[RULENAME,PREBIND]
calls: SWEEPER,SUBLIS,GETPROP,CONS
called by: STAMMER

ARRLOC[ARR]
calls: ARRAYP,FKARRAYP,IPLUS,LOC,ERROR

ASSERT[ARGLIST,NODENAME]
calls: SET,SAVEPULSAR,SERT,GETSTRIP,GENSYM,CONS
binds: REPLY,LEN,A
uses free: ASSERTIONS

ASSRPRINT[PRINSPEC]
calls: STRINGP,NUMBERP,PRIN1,NTH,EVAL,TERPRI
called by: PRETTYASSR
uses free: BODY,OVERCONF,LSTFLG

AUXINTERPOL[PT1,PT2,DELTA]
calls: LIST,FPLUS,FTIMES,FDIFFERENCE,FIXLONG
called by: ESTIMATE,ONEPOINT

BEARING[SITE]
calls: GETATT,OWNPOS,CENTROID,DIRECTION
binds: TIME,POS1,POS2

BEEP[]
calls: PRIN1,CHARACTER
called by: QHASK

BEYONDINTEREST[TXT]
called by: MSGMTR

BKDSPBUF[X]

calls: MAPC, CHCON, FKJSYS
 called by: DSPINIT
 binds: C
 uses free: DSPTTYCODE

BLFN[BNODE]

calls: GREATERP, GETMD, GETCON, GETMB, MIN
 called by: BMEAS, DMEAS
 binds: BNCON
 uses free: BMEASANS

BMEAS[BBOX]

calls: MINUSP, FGREATERP, RESETLST, GETPROP, RESETSAVE, MARKON, MAPC, BLFN,
 DLFN, FTIMES, FDIFFERENCE, FPLUS
 called by: GETMB
 binds: BMEASANS, RULECON, BASTLST, DMEASANS
 uses free: BAST, MBCOMB

BUMP[L]

calls: DREVERSE, CONS
 called by: SERT
 binds: ANS

CASSERT[SPEC, VAL]

calls: GREATERP, PUTPROP, SET, SAVEPULSAR, SERT, GETSTRIP, GENSYM, CONS, ABS
 called by: EWMSG, ORACLEHACK, OWNMSG, SENSORMSG, UNLESSHACK, WEATHERMSG, DENY,
 MAYBE, STATE
 binds: NEWNODE
 uses free: ASSERTIONS

CENTROID[VERTEXLIST]

calls: LIST, FQUOTIENT, ADD1, FPLUS
 called by: BEARING, COURSE, COURSEFROM, CROSSPATHS, DESCRIBMSG, GRAZE, IN-LANE,
 INSIDE, LOC-TIME, LOCATION, ONEPOINT, RANGE, SPEED, SPEEDFROM, SWR,
 WENT-AFTER, WENT-BEFORE
 binds: I, C1, C2

CHANGECON[RLNME1]

calls: PRIN1, PUTPROP, CLEARBUF, GETPROP, READ, TERPRI

CKCONFIGURATION[]

calls: TEKTEST, EQP, DSPINIT, PRIN1, DSPMAP, CLEARBUF, STARTUP, HELP, CHCON1,
 ASKUSER, DSPGRAB, READ, DSPTTYSTR
 called by: WELCOME
 uses free: DUALFLG, DISPLAYFLG

CONSTRUCT[ACTIONS, EV, COUNT]

calls: ORBUILD, JUSTBUILD, CONS, MESSAGE1
called by: ORBUILD, SWEEPER
binds: FIRST
uses free: RESULTLIST

COURSE[SITE]

calls: LESSP, GETATT, CENTROID, PREDECESSOR, SUCCESSOR, DIRECTION,
FDIFFERENCE
binds: TIME, POS, PRED, SUC, TSUC, PSUC, TPRED, PPRED

COURSEFROM[POS1, POS2]

calls: CENTROID, DIRECTION

CREATH[SIZE]

calls: ARRAY, ADD1, IQUOTIENT, LOG, ITIMES
called by: NEWHASH
uses free: MEMFULLSIZE, MEMFILLED, MEMFACTOR, MEMSIZE, MEMORY

CROSSBOUNDARY[PT1, PT2, POLY]

calls: SOMELINESEG, FUNCTION, CROSSLINES
called by: LINPOLY
binds: PT3, PT4

CROSSLINES[A, B, P, Q]

calls: OPSIDES
called by: CROSSPATHS, CROSSBOUNDARY

CROSSPATHS[S1, S2, T1, T2]

calls: CENTROID, CROSSLINES
binds: P1, P2, Q1, Q2

CRUNCH[X]

calls: RPTQ, SETN, LOGOR, LLSH
called by: DSPCNVRT
binds: NUM, RPTN

DECSAMEDIGITS[X]

calls: LESSP, IPLUS, IREMAINDER, ITIMES, DECSAMEDIGITS, IQUOTIENT
called by: DECSAMEDIGITS, DSPTTYSTR

DEFINEPD[]

calls: PRIN1, MAKEPD, LINEREAD, APPEND
binds: PDNAME, NEWCON, CONDS, NEWACT, ACT, CONFID

DENY[L]
calls: CASSERT

DESCRIBMSG[TXT]
calls: NUMBERP, TERPRI, SPACES, PRIN1, WAITER, DSPCMD, LAST, CENTROID, LIST
called by: MSGMTR
binds: POS, TIME, WKNM, SOURCE
uses free: OWNSHIP, CURTIME, DSPLAYFLG

DIRECTION[LAT1, LON1, LAT2, LON2]
calls: EQP, FGTP, LESSP, MINUSP, SETN, SUBTEND, FDIFFERENCE, FQUOTIENT,
FTIMES, COS, SIN, ARCSIN, FPLUS
called by: BEARING, COURSE, COURSEFROM, INTERIOR, ROTSENSE, WENT-AFTER,
WENT-BEFORE
binds: PSI, LONDIF, BEARSIN, BEARANGLE

DISPCHECK[NAME]
calls: GETPROP
called by: DISPLAY

DISPLAY[PLATNAME, LAT, LON, TIME]
calls: DISPCHECK, DSPADDTRH, DISPMARK, DSPADDINC, FLOAT, MELD, IDENT, MEDIUM
called by: ADDIS, DISPLOB, OWNMSG, SENSORMSG
uses free: DSPLAYFLG

DISPLOB[PNAME, SPOS, DPOS, TIME]
calls: DISPLAY, MIDP
called by: EWMSG
binds: TEMPl, TEMP2
uses free: DSPLAYFLG

DISPMARK[NAME]
calls: PUTPROP
called by: DISPLAY

DISSIMILPLAT[PLAT1, PLAT2]
calls: EQUAL, MAPC, GETATT
called by: POSS-REPORT
binds: VAL1, VAL2, SUCCESSFLG, CHAR
uses free: SHIPCHARS

DISTANCE[LAT1, LON1, LAT2, LON2]
calls: FTIMES, SUBTEND
called by: DISTOLINE, NEAREST, RANGE, SPEED, SPEEDFROM, SWR, WENT-AFTER,
WENT-BEFORE

DISTOLINE[X,Y,X1,Y1,X2,Y2]
 calls: MINUSP,SETN,DISTANCE,QUOTIENT,FDIFFERENCE,FPLUS,FTIMES,TIMES,
 MIN,SIN,ARCCOS
 called by: GRAZE,INLANE
 binds: A,B,C,COS1,COS2

DLFN[DNODE]
 calls: FGREATERP,GETMB,GETCON,GETMD,MAX
 called by: BMEAS,DMEAS
 binds: DNCON
 uses free: DMEASANS

DMEAS[DBOX]
 calls: MINUSP,FGREATERP,RESETLST,GETPROP,RESETSAVE,MARKON,MAPC,BLFN,
 DLFN,FTIMES,FDIFFERENCE,FPLUS
 called by: GETMD
 binds: DMEASANS,RULECON,DASTLST,BMEASANS
 uses free: DAST,MDCOMB

DSPADDINC[NAME,LAT,LON,TIME]
 calls: FKCALL,DSPCNVRT
 called by: DISPLAY,STUFFLN,WEATHERMSG

DSPADDINCS[NAME,INCLST]
 calls: FKCALL,DSPCNVRT,MAPC
 binds: INC

DSPADDTRH[NAME,ID,TYPE]
 calls: FKCALL,DSPCNVRT
 called by: DISPLAY,STUFFLN,WEATHERMSG

DSPCHGTRH[NAME,ID,TYPE]
 calls: DSPQUIET,DSPEXCH,FKSETVAL,FKCALL,DSPTTY

DSPCMD[CMD,WAITFLG]
 calls: EQP,FKCALL,GRATEK,TERPRI,GETTOPVAL,MONTAK
 called by: DESCRIBMSG,PRINTRULEASSR
 binds: DSPNOWAITFLG
 uses free: DSPTTYCODE,TEK4025

DSPCNVRT[X]
 calls: DCHCON,CRUNCH,NTH
 called by: DSPADDINC,DSPADDINCS,DSPADDTRH,DSPINIT
 uses free: SCRATCHTEN,DSPWORD1,DSPWORD2

AD-A084 053

SDC INTEGRATED SERVICES INC SAN DIEGO CA F/6 9/2
STAMMER2 PRODUCTION SYSTEM FOR TACTICAL SITUATION ASSESSMENT, V--ETC(U)
OCT 79 D C MCCALL, P H MORRIS, D F KIBLER N00123-76-C-0172

UNCLASSIFIED

NOSC-TD-298-VOL-2

NL

3 OF 3

NO
ACQUISITION



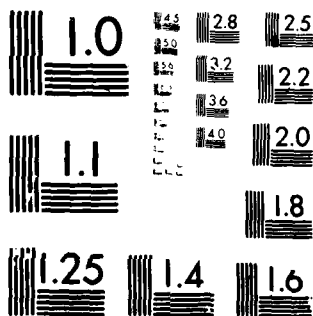
END

DATE

FILMED

6-80

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

DSPERASE[]

calls: FKCALL, TEKWAIT

DSPEXCH[NAME]

calls: FKCALL, RPTQ, RPLSTRING, ADD1, ITIMES, NTHCHAR
 called by: DSPCHGTRH
 binds: RPTN
 uses free: DSPEXCHBUF

DSPEXP[BOX]

calls: MEMB, ZEROP, IGREATERP, MAPC, CONS, GETUPLE, ADD1, APPLY, DREVERSE,
 CONCAT
 called by: PRINTRULEASSR
 binds: X, DSPLST, COUNT, DSPOBJECTS, BLANK, COMMA
 uses free: OWNSHIP

DSPGRAB[TTYNO]

calls: IPLUS, OCTSAMEDIGITS, FKJSYS
 called by: CKCONFIGURATION
 uses free: DSPTTYCODE

DSPINIT[]

calls: FKINIT, DSPQUIET, BKDSPBUF, DSPCNVRT, FKSETVAL, FKCALL, DSPTTY,
 DSPTTYSTR, TERPRI
 called by: CKCONFIGURATION
 globals: DSPNOWAITFLG

DSPMAP[]

calls: FKSETVAL, INFILEP, PRIN1, TERPRI
 called by: CKCONFIGURATION

DSPNOMAP[]

calls: FKSETVAL

DSPNUMAT[X]

calls: MAPCAR, DSPNUMAT, PACKC, UNCRUNCH
 called by: DSPNUMAT, FKVALAT

DSPQUIET[]

calls: FKCALL, FKJSYS, LOGOR, LLSH
 called by: DSPCHGTRH, DSPINIT
 uses free: DSPTTYCODE
 globals: FORKDATA

DSPRELD[]

calls: FKJSYS
 called by: DSPSAVE, PARTING
 uses free: DSPTTYCODE

DSPSAVE[]

calls: FKCALL, DSPRELD, FKKILL, FKSAVE

DSPSTAT[]

calls: FKJSYS, LRSH
 uses free: FKJSYSAC1
 globals: FORKDATA

DSPTOP[WAITFLG]

calls: FKCALL, EQP, RESETLST, GETTOPVAL, RESETSAVE, GRATEK
 binds: DSPNOWAITFLG, FIRSTCMD
 uses free: DSPTTYCODE, TEK4025

DSPTTY[]

calls: FKCALL, FKJSYS, LOGOR, LLSH
 called by: DSPINIT, DSPCHGTRH
 uses free: DSPTTYCODE
 globals: FORKDATA

DSPTTYSTR[]

calls: EQP, CONCAT, DECSAMEDIGITS, IDIFFERENCE
 called by: CKCONFIGURATION, DSPINIT
 uses free: DSPTTYCODE

ENDSTREAM[S]

calls: RPLACD
 called by: SERT

ESTIMATE[SITE1, SITE2, GAP]

calls: PRIN1, TERPRI, MAPCAR, SPAN, GETATT, AUXINTERPOL
 called by: PREDICTPOS
 binds: X
 uses free: EXPLAINFLAG

EWMSG[TXT, EXTFLG]

calls: CASSERT, DISPLOB, NEWSYM, LIST, GETPOINT, OWNPOS
 called by: MSGMTR
 binds: TIME, TEMPLAC1, TEMPLAC2, SNODE, SOURCE, BEAR, EMIT, WKNM
 uses free: SENSORANGE

EXLOOP[]
 calls: RESOUT, EXPLAIN, MSGMTR
 called by: STAMMER
 binds: MSGFLG

EXPLAIN[]
 calls: QHTAKE, MAPC, CONS, RETRIEVES, NEWVALOBJ, ERSETQ
 called by: EXLOOP
 binds: DONEFLG, PLATFORM, MLANE, STORM, ID, ID-AMP, TYPE, VALUE, OBJECT
 uses free: ASSERTIONS, RELATIONS, PRODUCTIONS, RELATION, RULENAME

ANCYPROD[PRO]
 calls: PRIN1, PRINT, TAB, GETPROP, PRETTYASSR, MAPC, TERPRI
 binds: C, A, X, CART, CARTEL

ASTHAK[]
 calls: MAPH, FUNCTION, MEMTEST
 uses free: MEMSIZE, MEMORY

IXLONG[X]
 calls: FLESSP, FDIFFERENCE, FPLUS
 called by: AUXINTERPOL, GETPOINT

KACS[]
 calls: RPLACA, ARRAY
 called by: FKCALL, FKSETVAL, FKVALI, SAILCALL
 binds: Y, X
 globals: FORKDATA

KACSRETURN[ARRAY]
 calls: RPLACA, CONS
 called by: FKCALL, FKSETVAL, FKVALI, SAILCALL
 binds: Y
 globals: FORKDATA

KARRADR[FKARRNAME, FKINDEX, FKNWORDS]
 calls: GETD, NUMBERP, ERROR, FKBCHECK, MAPC, EVAL, FKARRAYP, LOC, MAPCAR, LOGAND, OPENR, SUB1, LRSH, EQP, LENGTH, IDIFFERENCE, IPLUS, ITIMES, ADD1
 called by: FKELTI, FKELTR, FKSETA
 binds: FKADR, FKDIM, FKSIZE, FKOFFSET, FKNDIM, FKPTR, FKLOW, FKARRY, X

FKARRAY[KA,FKTYPE,FKSIZE,FKSIZE2]

calls: GETD,NUMBERP,ILESSP,NOFORK,FKSYMPUT,MAPC,EVAL,MAPCAR,ERROR,
 ADD1,ITIMES,LIST,REVERSE,CONS,IPLUS,IDIFFERENCE,FKCORGET,LOGOR,
 FKHT,FKSHR,IMINUS,FKIDPB,LLSH,SET,VAG
 binds: FKHI,FKOFFSET,FKNDIM,FKTOTALSIZE,FKDOPE,FKDIMS,FKLO,FKLOC,
 FKBYTP,FKDATAWD,FKSIZES,WORD
 globals: FORKDATA

FKARRAYP[A]

calls: IGREATERP,FKSHR,LOC
 called by: ARRLOC,FKCALL,SAILARG,FKARRADR
 binds: SHR
 globals: FORKDATA

FKARRAYSIZE[A]

calls: LOGAND,OPENR,SUB1,LOC

FKARRAYTYPE[A]

calls: ZEROP,LRSR,OPENR,SUB1,LOC,IDIFFERENCE,IPLUS,ITIMES
 called by: FKCALL,FKELT
 binds: NDIM

FKBCHECK[N,LO,HI]

calls: IGREATERP,ILESSP,ERROR
 called by: FKARRADR

FKCALL[FKCX]

calls: LITATOM,FXP,FLOATP,STRINGP,ARRAYP,IGREATERP,FKARRAYP,NOFORK,
 FKWAIT,FKIDPB,FKCALLERR,RPTQ,SETA,ERROR,PUTTYP,FKSACS,FKSW,
 FKACSRETURN,FKRACS,MAPC,FKHNDL,FKHT,FKACS,LOGOR,IPLUS,LOC,EVAL,
 FKSYP,FKCATYPE,SUB1,CONS,FKSR,ARRAYSIZE,IDIFFERENCE,ELT,
 FKARRAYTYPE,FKSHR,ADD1,SET,FKRTN
 called by: DSPADDINC,DSPADDINCS,DSPADDTRH,DSPEXCH,DSPQUIET,DSPTOP,DSPTTY,
 DSPCHGTRH,DSPCMD,DSPERASE,DSPINIT,DSPSAVE
 binds: FKHNDL,FKHT,FKCA,FKCBP,FKCABP,FKCID,FKCARG,FKBIAS,FKCTYPE,
 FKRESLIST,FKCWRDS,FKCN,FKCARGS,FKRESULTYPE,FKRESULT,RPTN,X,
 FKVAL
 globals: FORKDATA,DSPNOWAITFLG

FKCALLERR[FKCID]

calls: ERROR
 called by: FKCALL,SAILCALL

FKCATYPE[FKID]

calls: ILESSP,IGREATERP,CHCON1
 called by: FKCALL,FKVAL
 binds: C

FKCORGET[SIZE]

calls: IGREATERP, ERROR, RPLACA, FKSHR, FKPROG, IDIFFERENCE
 called by: FKARRAY
 binds: SHR, X
 globals: FORKDATA

FKDDT[DDTFILE]

calls: ZEROP, EQP, NOFORK, ERROR, FKDDT_, RESETFORM, FKHNDL, FKDDT, LOGOR,
 LLSH, LOGAND, FKJSYS, MKSTRING, FKSW, FKTTYSET
 called by: FKDDT, FKKILL
 binds: FKHNDL, EV, DDT
 uses free: FKJSYSAC1, FKJSYSAC2
 globals: FORKDATA

FKELT[FKELT!A, FKELT!N, FKELT!WORDS]

calls: APPLY*, FKARRAYTYPE, EVAL, FUNCTION, FKELTR, FKELTI

FKELTI[FKELTI!A, FKELTI!N, FKELTI!WORDS]

calls: EVAL, FKARRADR, RPTQ, CONS, OPENR, IPLUS
 called by: FKELT
 binds: PTR, ANS, RPTN

FKELTR[FKELTR!A, FKELTR!N, FKELTR!WORDS]

calls: EVAL, FKARRADR, RPTQ, CONS, FKFLOAT, IPLUS
 called by: FKELT
 binds: PTR, ANS, RPTN

FKFLOAT[ADR]

calls: ASSEMBLE, VAG
 called by: FKELTR, FKVALI

FKINIT[PROGRAM]

calls: MEMBER, ILESSP, IGREATERP, FKACRS, RPTQ, ARRAY, HARRAY, INFILEP,
 UNPACK, PACK, LIST, ERROR, FKJSYS, MKSTRING, LOGOR, LLSH, LRSH,
 RESETFORM, FKTTYSET, FKSW, ADD1, ELT, EQP, LOC, GETBLK, SUB1, IPLUS,
 IDIFFERENCE
 called by: DSPINIT, NOFORK
 binds: PROGFIL, FKJFN, FKHNDL, HALTED, EV, HALTADR, PGS, SOURCE, DEST, FKSHR,
 FKACS, LISPBLOCK, RPTN, SIZE, FKHT, FKSVMACS, FKDDT
 uses free: FKJSYSAC2, FKJSYSAC1
 globals: FORKDATA

FKJSYS[FKJSYSNO, ARG1, ARG2, ARG3, ARG4, ARG5]

calls: ASSEMBLE, VAG, FKJSYSARG, LOC, AC

called by: FKTIME, DSPQUIET, DSPSTAT, DSPTTY, FKKILL, FKSR, FKSW, FKTTYSET, FKWAIT, BKDSPBUF, DSPGRAB, DSPRELD, FKDDT, FKINIT, FKSAVE

uses free: FKJSYSAC3, FKJSYSAC2, FKJSYSAC1

FKJSYSARG[X]

calls: STRINGP, ZEROP, ARRAYP, NUMBERP, CHCON1, NTHCHAR, CONCAT, CHARACTER, IPLUS, LOC, LSH, LOGAND, LOGOR, IQUOTIENT, LLSH, IDIFFERENCE, ITIMES, IREMAINDER, ERROR, VAG

called by: FKJSYS

binds: ARG, S

uses free: FKJSYSTR

FKKILL[]

calls: EQP, FKJSYS, RELBLK, EVALV, FKHNDL, RSH, FKPROG, FKJFN, FKDDT, FKSHR, VAG, LRSH, IDIFFERENCE

called by: DSPSAVE, PARTING

binds: FKPROG, DDT, SHR

uses free: FKJSYSAC1

globals: FORKDATA

FKRTN[TYPE, A, N]

calls: ZEROP, ELT, ASSEMBLE, VAG, IPLUS, LOC, ERROR

called by: FKCALL, SAILCALL

FKSAVE[FILE]

calls: ERROR, FKJSYS, MKSTRING, LOGOR, LLSH, FKHNDL, INFILEP

called by: DSPSAVE

binds: JFN

uses free: FKJSYSAC1

globals: FORKDATA

FKSETA[FKARRY, FKINDEX, FKEXPR]

calls: EVAL, FKARRADR, LENGTH, MAPCAR, CLOSER, ADD1

binds: FKVAL, FKPTR, FKV

FKSETVAL[FKADR, FKBIAS, FKVAL]

calls: IGREATERP, NUMBERP, NOFORK, FKWAIT, HELP, RPLACD, MAPC, FKSACS, FKSW, FKACSRETURN, FKHNDL, FKHT, FKACS, LOGOR, IPLUS, LOC, FKSYM, EVAL, LIST, LENGTH, COPY, NTH, FKIDPB, ERROR

called by: DSPMAP, DSPNOMAP, DSPCHGTRH, DSPINIT

binds: VAL, FKHNDL, FKHT, FKACS, FKBP, FKRESULT

globals: FORKDATA

FKSR[A, I, STR]

calls: IGREATERP, RPTQ, SETA, FKJSYS, LOGOR, IPLUS, LOC, NCHARS, IDIFFERENCE,
ARRAYSIZE, SUB1, ADD1, IQUOTIENT, IMINUS
called by: FKCALL
binds: WDS, SIZE, ROOM, RPTN, DESTPTR

FKSW[FKHNDL, I, FKNOWAITFLG]

calls: ILESSP, EQP, FKJSYS, RESETFORM, FKHALT, LOGAND, RADIX, HELP
called by: FKCALL, FKSETVAL, FKSYP, FKVALI, SAILCALL, FKDDT, FKINIT
binds: EXPECTED, HALTED
uses free: FKJSYSAC2
globals: FORKDATA, DSPNOWAITFLG

FKSYM[ID, FKHT, NOBREAK]

calls: ZEROP, SETA, FKSACS, FKSW, FKRACS, ERROR, FKHT, NOFORK, FIXP, GETHASH,
FKSYMACS, FKHNDL, GETRADIX50, ELT, FKSYPUT
called by: FKCALL, FKSETVAL, FKSYP, FKVALI, SAILARG, SAILCALL
binds: P, FKHNDL
globals: FORKDATA

FKSYP[ID]

calls: FKSYP

FKSYPUT[FKHT, ID, V]

calls: PUTHASH, FKHT_, LIST
called by: FKARRAY, FKSYP
binds: HTL
globals: FORKDATA

FKTIME[FKEXPR]

calls: FKJSYS, FKHNDL, EVAL, FQUOTIENT, IDIFFERENCE, LIST, FPLUS
binds: FKHNDL, FKFORKTIME, FKLISPTIME, FKRESULT
uses free: FKJSYSAC2, FKJSYSAC1
globals: FORKDATA

FKTTYSET[BOOL]

calls: FKJSYS
called by: FKDDT, FKINIT
uses free: FKJSYSAC3, FKJSYSAC2, FKTTYSETCALLED, FKTIW, FKFMOD, FKCC2, FKCC1

FKVAL[FKADR, FKBIAS, FKWORDS]

calls: APPLY*, FKCATYPE, FKVALI

FKVALAT[ID, BIAS, NVALS]

calls: DSPNUMAT, APPLY, LIST, FKVALI

FKVALI[FKADR,FKBIAS,FKWORDS,FKREAL]

calls: IGREATERP,NOFORK,FKWAIT,FKIDPB,FKSACS,FKSW,FKRACS,HELP,RPTQ,
FKACSRETURN,FKHNDL,FKHT,FKACS,LOGOR,IPLUS,LOC,FKSYM,EVAL,CONS,
FKFLOAT,OPENR

called by: FKVAL,FKVALAT,FKVALR

binds: FKHNDL,FKHT,FKACS,FKBP,FKRESULT,RPTN

globals: FORKDATA

FKVALR[FKADR,FKBIAS,FKWORDS]

calls: APPLY*,FKVALI

FKWAIT[FKHNDL]

calls: MEMB,FKJSYS,DISMISS,HELP,LRSR,LOGAND

called by: FKCALL,FKSETVAL,FKVALI

uses free: FKJSYSAC2,FKJSYSAC1,FKSTATUS

FKX[FKCX]

calls: EVAL,LIST

FREEZE[]

called by: MSGMTR

uses free: FREEZEFLG

GAMF[WLK,OVERRIDE]

calls: FGREATERP,FLESSP,PRIN1,GETCON,ABS,EQP

called by: MODIFIER,NICEANSWER,YESNO

binds: CONF1,ACON

GETATT[REL,NAME]

calls: RPLACA,GETUPLE,STRIPSTREAM,GETSH

called by: ADDIS,BEARING,COURSE,DISSIMILPLAT,ESTIMATE,LOC-TIME,LOCATION,
ONEPOINT,PLATPOS,PREDCESSOR,PREDICTPOS,RANGE,SPEED,STUFFLN,
SUCCESSOR

binds: SPEC

GETATTB[REL,NODE]

calls: RPLACA,GETUPLE,STRIPSTREAM,GETSH

called by: ADDIS,POSS-REPORT,PREDCESSOR,SUCCESSOR

binds: SPEC

GETCON[SOMAST]

calls: ATOM,FDIFFERENCE,GETMB,GETMD,MAPCAR,GETCON

called by: BLFN,DLFN,GAMF,GETCON,GREATESTPROB,IMPLIESASRT,MODIFIER,
PRINTRULEASSR,YESNO,ANDHACK,NOTHACK,UNLESSHACK

GETH[ARGS]

calls: ELTD, LOCH
 called by: GETSH, GETSTRIP
 uses free: MEMORY

GETMARK[NODE]

calls: GETPROP
 called by: GETMB, GETMD

GETMB[BAST]

calls: MAPC, GETPROP, GETMARK, BMEAS
 called by: BLFN, DLFN, GETCON
 binds: HNDL, MBCOMB

GETMD[DAST]

calls: MAPC, GETPROP, GETMARK, DMEAS
 called by: BLFN, DLFN, GETCON
 binds: DNDL, MDCOMB

GETMRVAL[X, COPYFLG]

calls: SUBLIS
 called by: MESSAGE1, ORACLEHACK, SWEEPER
 uses free: MAPRETALIST

GETPOINT[POS, BEAR, RANGE]

calls: EQUAL, FGTP, FQUOTIENT, SIN, COS, ARCSIN, FPLUS, FTIMES, ABS, LIST,
 ARCCOS, MAX, MIN, FDIFFERENCE, FMINUS, FIXLONG
 called by: EWMSG
 binds: SINLAT, COSPSI, COSLAT, SINPSI, COSBEAR, NEWLAT, COSNEWLAT, TMP, TMP2,
 NEWLONG, LAT, PSI, LONG

GETPULSAR[NODE]

calls: GETPROP
 called by: JUSTBUILD, MAPRETDO

GETRADIX50[S]

calls: ILESSP, IGREATERP, RPTQ, NCHARS, SUBSTRING, CHCON1, GNC, IDIFFERENCE,
 IPLUS, ITIMES
 called by: FKSVM
 binds: RADTMP, RAD, LEN, TS, RPTN

GETSH[ARGS]

calls: GETH, PUTH, APPEND, NEWSTREAM
 called by: GETATT, GETATTB, PUTSH, RETRIEVES, RETSTREAM, SERT

GETSTRIP[ARGS]

calls: STRIPSTREAM,GETH
called by: ASSERT,CASSERT,JUSTBUILD,RETRIEVER,YESNO

GETUPLE[ASSER]

calls: EVAL
called by: DSPEXP,GETATT,GETATTB,IMPLIESASRT,NEWVALOBJ,RESULTPRINTER,
RETPULSED0,RETRIEVER,RETRIEVES

GRATEK[]

calls: TEKCOM,PRINCHAR,PRIN1,JSYS,TERPRI,DOBE
called by: DSPCMD,DSPTOP

GRAZE[S1,S2,T1,T2]

calls: CENTROID,LESSP,DISTOLINE
binds: POS1,POS3,POS4,POS2
uses free: PATROLRANGE

GREATER-THAN[Q1,Q2]

calls: GREATERP

GREATESTPROB[POSLIST]

calls: GREATERP,EQP,MAPC,GETCON
called by: IDENT,MEDIUM
binds: ANSCON,ANS,A

HLPEXPLN[]

calls: PRIN1,TERPRI

IDENT[NAME]

calls: RETRIEVER,LIST,GREATESTPROB
called by: DISPLAY
binds: POSIB,ANS

IMPLIESASRT[NODE]

calls: EQP,TERPRI,PRIN1,MAPC,GETPROP,GETCON,GETUPLE,MEMBER,APPEND,
LIST,SPACES
binds: X,Y
uses free: RULE

IN-LANE[MLANE,POS]

calls: CENTROID,LAST,FGREATERP,LANERANGE
binds: X,Y

INCLUDEPLAT[PNE]

calls: MAPC, RETRIEVES, ADDIS
called by: STARTUP

INLANE[X,Y,LANE]

calls: SOME, SETN, LESSP, DISTOLINE
binds: X1, Y1, X2, Y2, LANEPOINT
uses free: MERCHANTLANEWIDTH

INSIDE[POS, STORM]

calls: APPLY, APPEND, CENTROID, CONS, INTERIOR

INTERIOR[OLAT, OLON, POLYGON]

calls: LESSP, SETN, MAPC, LAST, DIFFERENCE, DIRECTION, PLUS, GREATERP, ABS
called by: LINPOLY, INSIDE
binds: SUM, INC, POS1, POS, LAT, LON, LAT1, LON1

INTERPOLABLE[TXT]

called by: MSGMTR

JUGGLE[PAIR, INSERTITEM]

calls: LIST

JUSTBUILD[SPEC, EV, NUMBER]

calls: PUTPROP, SET, SAVEPULSAR, SERT, PULSE, MASSAGE1, GETSTRIP, GENSYM,
CONS, REVERSE, GETPROP, GETPULSAR
called by: CONSTRUCT
binds: MESSAGESPEC, NEWNODE, NEWFLG
uses free: ASSERTIONS, RESULTLIST

LANERANGE[ALAT, ALON, BLAT, BLON, CLAT, CLON]

calls: SETN, COS, SIN, FTIMES, ABS, FDIFFERENCE, ARCCOS, FQUOTIENT, FPLUS,
SUBTEND
called by: IN-LANE
binds: A1, B1, C1, A2, B2, C2, A3, B3, C3, CAT, CAN, CBT, CBN, CCT, CCN, SAN, SBN, SCN,
SAT, SBT, SCT

LESS-THAN[Q1, Q2]

calls: GREATERP

LINEREAD[]

calls: BKLINBUF, READP, READLINE
called by: DEFINEPD, WELCOME

LINPOLY[PT1, PT2, POLY]

calls: CROSSBOUNDARY, INTERIOR
called by: TRACKINPOLY

LOC-TIME[S]

calls: NCONC1, CENTROID, GETATT

LOCATION[S]

calls: CENTROID, GETATT

LOCH[ARGS, PUTFLG]

calls: EQUAL, PREHASH, ELT, NEXTH, ADD1

called by: GETH, PUTH

binds: LOC, CONT

uses free: MEMORY, MEMTESTCNT

M[L]

calls: NCONC, MAKEFILE

uses free: DSPLAFNS

MAKEPD[NAM, CO, AC, TRUST]

calls: PUTPROP, CONS

called by: DEFINEPD

uses free: PRODUCTIONS

MAKEPRINT[RELN]

calls: TERPRI, MAPC, SPACES, PUTPROP, GETPROP, PRIN1, READ, APPEND, CONS

binds: PFORM, NEWFORM, X

MAPH[ARY, ARYSZ, ARYFN]

calls: GREATERP, APPLY*, ELT, ELTD, ADD1

called by: NEWHASH, FASTHAK

binds: COUNT, CONTENT

MAPRETD0[SELT, AI]

calls: SOMEPUSE, GETPULSAR, CONS, FUNCTION, RETPULSED0

called by: MAPRETRIEVE

MAPRETRIEVE[MAPRETX, MAPRETINFO, MAPRETFN]

calls: MAPSTREAM, RETSTREAM, CONS, FUNCTION, MAPRETD0

called by: ANDHACK, NOTHACK, UNLESSHACK

uses free: MAPRETALIST

MAPSTREAM[MAPSTREAMX, MAPSTREAMINFO, MAPSTREAMFN]

calls: MAPC, TCONC, APPLY*, CONS

called by: MAPRETRIEVE

binds: X

MARKOFF[NODE]

calls: REMPROP

MARKON[NODE, MARK]

calls: PUTPROP

called by: BMEAS, DMEAS

MASSAGE[LIST]

calls: VAR?, MAPCAR, GETMRVAL

called by: CONSTRUCT, JUSTBUILD, UNLESSHACK

binds: X

MATCHER[L1, L2]

calls: DREVERSE, CONS

called by: SERT

binds: ANS

MAYBE[L]

calls: CASSERT

MEDIUM[NAME]

calls: RETRIEVER, LIST, GREATESTPROB

called by: DISPLAY

binds: TEMP1, RETURNER

MELD[ID, MED]

calls: PACK, LIST

called by: DISPLAY

binds: A, B

MEMDENSITY[]

calls: PRIN1, FTIMES, FQUOTIENT, TERPRI

uses free: MEMSIZE, MEMFILLED

MEMSAVE[FEE]

calls: SET, MAKEFILE, TERPRI, PRIN1, FILECOMS, CLEARBUF

uses free: MEMORYCOMS

MEMTEST[X, Y]

calls: PRIN1, LENGTH, TERPRI

called by: FASTHAK

uses free: COUNT

MIDP[P1, P2]

calls: GREATERP, FDIFFERENCE, ABS, MINUS, TWO-PLACE, FQUOTIENT, FPLUS

called by: DISPLOB

binds: TEMP2

MODIFIER[]

calls: MEMBER, GAMF, PRIN1, GETCON, TWO-PLACE
 binds: CON
 uses free: NODE, OVERCONF

MONTEK[]

calls: TEKCOM
 called by: DSPCMD

MSGMTR[]

calls: BEYONDINTEREST, INTERPOLABLE, NUMBERP, INFILE, FREEZE, CLOSEF?,
 UNFREEZE, OWNMSG, DESCRIBMSG, PRIN1, TERPRI, WEATHERMSG, SENSORMSG,
 EWMSG, INPUT, READ
 called by: EXLOOP
 binds: OLDIN, MSG
 uses free: OWNSHIP, MSGFILE

NEAREST[PT, LST]

calls: FLESSP, MAPC, DISTANCE
 called by: SPAN
 binds: TEMP, Y, ANS, X

NEWHASH[]

calls: CREATH, MAPH, PLUS, IQUOTIENT, FUNCTION, PUTH
 called by: PUTH
 binds: LEFT, A, OLDSIZE, RIGHT
 uses free: MEMSIZE, MEMORY

NEWSTREAM[]

calls: CONS
 called by: GETSH

NEWSYM[NAME]

calls: PUTPROP, ADD1, GETPROP, CONS, PACK, APPEND, UNPACK
 called by: EWMSG, OWNMSG, SENSORMSG
 uses free: SYMBOLS

NEWVALOBJ[ARRT]

calls: LESSP, LENGTH, GETUPLE, NUMBERP, MEMB, CONS
 called by: EXPLAIN
 binds: TUPLE, VL, OJ
 uses free: OBJECT, VALUE

NEXTH[LOC,ARG]

calls: GREATERP, IDIFFERENCE, IPLUS
 called by: LOCH
 binds: NEWLOC
 uses free: MEMSIZE

NICEANSWER[ANS]

calls: GAMF, PRIN1, TERPRI
 called by: PRETTYANS

NOFORK[]

calls: PRIN1, APPLY*, READ, FKINIT
 called by: FKARRAY, FKCALL, FKDDT, FKSETVAL, FKVAL1, FKSVM, SAILCALL
 globals: FORKDATA

NOTHACK[CONDITIONS,ACTIONS,EV]

calls: ORACLEHACK, MAPRETRIEVE, FUNCTION, LESSP, GETCON, SWEEPER, CONS, LIST
 called by: SWEEPER
 binds: P, X, CLIST

OCCURPRINT[TIMES, NODE]

calls: ZEROP, EQUAL, PRINTRULEASSR, QHTAKE, GETPROP, SUB1
 binds: X, Y, Z
 uses free: RULE

OCTSAMEDIGITS[X]

calls: LESSP, IPLUS, IREMAINDER, ITIMES, OCTSAMEDIGITS, IQUOTIENT
 called by: DSPGRAB, OCTSAMEDIGITS

ONEPOINT[NODE,GAP]

calls: PRIN1, TERPRI, FTIMES, GETATT, LIST, FDIFFERENCE, FPLUS, MAPCAR, SPAN,
 CENTROID, AUXINTERPOL
 called by: PREDICTPOS
 binds: LAT, LONG, X, POS, Y
 uses free: EXPLAINFLAG

OPSIDES[A,B,P,Q]

calls: ROTSENSE
 called by: CROSSLINES

ORACLEHACK[SPEC]

calls: VAR?, EQUAL, RPLACD, NCONC, CASSERT, GETMRVAL, GETPROP, LAST, NLEFT,
 APPLY, RPLACA
 called by: ANDHACK, NOTHACK, UNLESSHACK
 binds: ORTYPE, LASTCONS, PTR, LAST-ARG, ANS

ORBUILD[SPEC, EV]
 calls: CONSTRUCT, LENGTH
 called by: CONSTRUCT
 binds: COUNT

ORHACK[CONDITIONS, ACTIONS, EV]
 calls: MAPC, SWEEPER, CONS
 called by: SWEEPER
 binds: TEMP2

OWNMSG[TXT]
 calls: CASSERT, DISPLAY, NEWSYM, LIST
 called by: MSGMTR
 binds: SNODE
 uses free: OWNSHIP

OWNPOS[TIME]
 calls: PLATPOS
 called by: BEARING, EWMSG, RANGE
 uses free: OWNSHIP

PARTING[]
 calls: PRIN1, TERPRI, FKKILL, DSPRELD
 called by: STAMMER
 uses free: DISPLAYFLG

PLATPOS[PLAT, TIME]
 calls: PRIN1, TERPRI, MAPCAR, RETRIEVER, LIST, CDADR, SUBSET, EQUAL, GETATT, PREDICTPOS
 called by: OWNPOS
 binds: X, Y, Z
 uses free: EXPLAINFLAG

POSS-REPORT[S1, S2, PATROL]
 calls: DISSIMILPLAT, MAPC, GETATTB
 binds: PLAT1, SUCCESSFLG, SNG, PLAT2

PQ[L]
 calls: SHOWPRINT, GETPROP
 binds: SYSPRETTYFLG

PREDECESSOR[SITE]
 calls: LESSP, MAPC, GETATTB, GETATT, RETRIEVES
 called by: COURSE, SPEED
 binds: PLAT, TOSSITE, TOSX, PRED, TOSPRED, X

PREDICTPOS[NODELIST,TIME]

calls: FLESSP,MAPC,GETATT,ESTIMATE,FQUOTIENT,FDIFFERENCE,ONEPOINT
 called by: PLATPOS
 binds: XT,LB2,LBT2,LB,LBT,UB2,UB,UBT,X
 uses free: UBT2

PREHASH[L]

calls: LITATOM,NUMBERP,STRINGP,ZEROP,SETN,ADD1,Iremainder,IPLUS,LSH,
 LOGAND,LOC,VAG,MKATOM,PREHASH,HELP,SUB1
 called by: LOCH,PREHASH
 binds: N,C
 uses free: MEMSIZE,PREHASHSUM1,PREHASHSUM

PREPALIST[CON,ASS,ALIST]

calls: VAR?,ASSOC,CONS
 called by: RETPULSED0
 binds: C,A

PRETTYANS[ANSLST]

calls: PRIN1,TERPRI,MAPC,NICEANSWER

PRETTYASSR[NODE,FORMAT,OVERCONF]

calls: GREATERP,PRIN1,FRPLACD,MAPC,MEMB,CONS,EVAL,GETPROP,LENGTH,LIST,
 PLUS,NTH,ASSRPRINT
 called by: FANCYPROD,PRINTRULEASSR,RECAPCONCS,RESULTPRINTER
 binds: LSTFLG,BODY,FORMLST,USEFORM,I,\$\$END
 uses free: ASSERTION

PRINCHAR[CODE]

calls: RESETFORM,ECHOCONTROL,PRIN1,CHARACTER
 called by: GRATEK,TEKTEST
 binds: X

PRINTRULEASSR[RULEASSRTS]

calls: ATOM,FLESSP,PRIN1,MAPC,TAB,PRETTASSR,WAITER,GETCON,PRETTYASSR,
 TERPRI,DSPCMD,DSPEXP
 called by: OCCURPRINT,RULEXP
 binds: Y
 uses free: DISPLAYFLG

PULSAR[]

calls: CONS
 called by: SAVEPULSAR

PULSE[PULSAR]

calls: APPLY*, RPLACA, CONSTANT, CONS, RPLACD
 called by: JUSTBUILD
 binds: CELL, PTR

PUTH[ARGS, AVAL]

calls: IGREATERP, NEWHASH, SETA, SETD, ADD1, LOCH
 called by: GETSH, NEWHASH
 binds: LOC
 uses free: MEMORY, MEMFULLSIZE, MEMFILLED

PUTSH[ARGS, AVAL]

calls: PUTSTREAM, GETSH
 called by: ADDH

PUTSTREAM[S, X]

calls: HELP, TCONC, MAPC, CONS, APPLY*
 called by: PUTSH
 binds: SUSP
 uses free: FREEZELST, FREEZEFLG

QHASK[INBUF]

calls: MEMB, NUMBERP, TERPRI, TCONC, BEEP, QHLIST, MAPRINT, MAPC, RESETLST, RESETSAVE, CONTROL, ECHOMODE, RAISE, ECHOCONTROL, QHGET, PEEKC, CONS, PRIN1, READC, CHCON1, GETSYNTAX, CHARACTER, MKATOM, CONCAT, SUBSTRING, RESETFORM, READ
 called by: QHFOLLOW
 binds: PTR, BUFPTR, CHAR, CODE, NEWPTR, ITEM, NUM, X

QHCLEAR[]

calls: CLRHASH
 called by: QHFOLLOW
 uses free: QUERYHASHPTR

QHFOLLOW[LL, BUFPTR, QHMATCH]

calls: STRINGP, TCONC, QHCLEAR, QHPREP, QHMAKE, TERPRI, QHSHOW, MAPRINT, CLEARBUF, QHFOLLOW, EVALA, CONS, PRIN1, NTHCHAR, CONCAT, CHARACTER, SUBSTRING, QHASK, READ, EVALV, EVAL, REVERSE
 called by: QHFOLLOW, QHTAKE
 binds: L, X, ALIST, QHVAL
 uses free: QHVAR

QHLIST[PTR]

calls: PRIN1, TERPRI, PRINT, RPTQ, QHLIST, QHGET, NTHCHAR, SUBSTRING, IDIFFERENCE
 called by: QHASK, QHLIST
 binds: ITEM, RPTN

QHMAKE[QHMAKEX, QHMAKEY, SHOWFLG]

calls: MEMB, MAPC, PRIN1, TERPRI, PRINT, RPTQ, QHPUT, QHMAKE, NTHCHAR,
 SUBSTRING, EVAL, MKATOM, NCHARS, CHCON1, IMINUS, QHGET, ADD1
 called by: QHFOLLOW, QHMAKE, QHPREP
 binds: CHARCODE, NEWPTR, PTR, X, RPTN
 uses free: QUERYHASHPTR

QHPREP[FOCUS, QHLST, SHOWFLG, STK]

calls: QHPREP, MAPC, NTHCHAR, GETPROP, HELP, CONS, QHMAKE
 called by: QHFOLLOW, QHPREP, QHSHOW
 binds: F, X

QHSHOW[L]

calls: QHPREP
 called by: QHFOLLOW

QHTAKE[L]

calls: QHFOLLOW, CONS
 called by: EXPLAIN, OCCURPRINT, RULEXP

RANGE[SITE]

calls: GETATT, OWNPOS, CENTROID, DISTANCE
 binds: TIME, POS1, POS2

RECAPCONCS[]

calls: TERPRI, MAPC, PRETTYASSR
 uses free: ASSERTION

RESOUT[]

calls: MAPC, INTERSECTION, RESULTPRINTER
 called by: EXLOOP
 uses free: RESULTLIST

RESULTPRINTER[RES1]

calls: MEMB, MAPC, PRIN1, TERPRI, GETUPLE, PRETTYASSR
 called by: RESOUT
 uses free: DULLREL

RETPULSED0[SELTAI]

calls: DECLARE, GETUPLE, PREPALIST, APPLY*
 called by: MAPRETDO
 binds: ASS, MAPRETALIST, MAPRETINFO, MAPRETX, MAPRETFN, SELT, AI

RETRIEVER[SPEC]

calls: MAPC, MAP2C, GETSTRIP, RETVARS, GETUPLE, VAR?, CONS
 called by: IDENT, MEDIUM, PLATPOS, WHAT2FORMFN, WHATFORMFN, WHOSE2FORMFN
 binds: RES1, RES, W, A, B

RETRIEVES[AT, OBJ, VAL, SEL]

calls: ILESSP, MEMBER, RPLACA, RPLACD, MAPC, NCONC, LAST, STRIPSTREAM, GETSH,
 ADD1, MAPCAR, NTH, GETUPLE, CONS
 called by: EXPLAIN, INCLUDEPLAT, PREDECESSOR, STARTUP, SUCCESSOR
 binds: SPEC, LAST, ASSES, X, ANS, ELT, ASS, ONEFLG

RETSTREAM[C]

calls: VAR?, MAPC, RPLACA, RPLACD, ASSOC, GETSH
 called by: MAPRETRIEVE, UNLESSHACK
 binds: PTR, XASSOC, FOLLOW, S, SCRATCH, X
 uses free: MAPRETALIST

RETVARS[SPEC]

calls: VAR?, MAPCAR
 called by: RETRIEVER
 binds: ITEM

ROTSENSE[A, B, C]

calls: LESSP, GREATERP, MINUSP, DIFFERENCE, DIRECTION
 called by: OPSIDES
 binds: ANGLE

ROUGHLY-THE-SAME-COURSE-AS [Q1, Q2]

calls: PLUS, GREATERP, DIFFERENCE

ROUGHLY-THE-SAME-SPEED-AS [Q1, Q2]

calls: PLUS, TIMES, GREATERP, DIFFERENCE

RULEXP[RULE, NODE]

calls: ZEROP, EQUAL, PRIN1, TERPRI, PRINTRULEASSR, QHTAKE, GETPROP, SUB1
 binds: X, COUNT, Y, Z

SAILARG[FKARG, FKHT]

calls: ATOM, FMEMB, LITATOM, STRINGP, FKARRAYP, FIXP, FLOATP, MAP, LAST, LIST,
 EVAL, FKSVM, SAILSTRING, IPLUS, LOC, FKSHR, ERROR, LLSH, FIX, FLOAT,
 LOGOR
 called by: SAILCALL
 binds: FKVARBL, FKVALUE, VARTYPE, FKTYPE, FKCALLTYPE, FKRV, FKARRY,
 FKRESULTS, X
 globals: FORKDATA

SAILARRAYSIZE[A]

calls: RPTQ, LOC, LRSH, OPENR, SUB1, IDIFFERENCE, CONS, REVERSE
 binds: NDIM, X, ANS, RPTN

SAILCALL[FKCX]

calls: ATOM, IGREATERP, ERROR, SETA, FKCALLERR, FKIDPB, MAPC, PUTTYP, FKSACS,
 FKSW, FKACSRETURN, FKRACS, FKHNDL, NOFORK, FKHT, FKACS, LOGOR, IPLUS,
 LOC, FMEMB, FKSVM, SAILARG, CONS, ADD1, LENGTH, SET, FKRTN
 binds: FKHNDL, FKHT, FKCA, FKCBP, FKCBP, FKRESULTBITS, FKARG, FKTYPE,
 FKRESLIST, FKCN, FKCARGS, FKRESULT, FKRESULTTYPE, FKCID, WORD, X
 uses free: FKTTYSETCALLED
 globals: FORKDATA

SAILSTRING[STRING]

calls: RPTQ, CHCON, CONS, REVERSE, IPLUS, LLSH
 called by: SAILARG
 binds: CHLIST, PACKEDLIST, VAL, ZEROS, RPTN

SAME-AS[W,U]**SAVEPULSAR[NODE]**

calls: PUTPROP, PULSAR
 called by: ASSERT, CASSERT, JUSTBUILD

SENSORMSG[TXT]

calls: EQUAL, CASSERT, DISPLAY, NEWSYM, LAST, LIST
 called by: MSGMTR
 binds: SNODE, LAT, LON, SOURCE, TIME, STR, WKNM

SERT[SPEC, NODENAME]

calls: MEMB, RPTQ, ADDH, ENDSTREAM, SUB1, LENGTH, CONS, MATCHER, BUMP, GETSH
 called by: ASSERT, CASSERT, JUSTBUILD
 binds: LEN, A, RPTN

SOMELINESEG[SOMELINESEGX, SOMELINESEGFN]

calls: SOME, APPLY*
 called by: CROSSBOUNDARY, TRACKINPOLY
 binds: SOMELINESEGPT1, SOMELINESEGPT2

SOMEPUULSE[PULSAR, PULSARDATA, SOMEPUULSEFN]

calls: APPLY*, TCONC, CONS
 called by: MAPRETDO

SPAN[L1,L2]

calls: IGREATERP, LENGTH, MAPCAR, LIST, NEAREST
called by: ESTIMATE, ONEPOINT
binds: X

SPEED[SITE]

calls: LESSP, GETATT, CENTROID, PREDECESSOR, SUCCESSOR, SPEEDM, DISTANCE, FDIFFERENCE
binds: TIME, POS, PRED, SUC, TSUC, PSUC, TPRED, PPRED

SPEEDAUX[T1,T2,DIST]

calls: FQUOTIENT, FDIFFERENCE
called by: SPEEDM

SPEEDFROM[POS1,T1,POS2,T2]

calls: CENTROID, SPEEDM, DISTANCE

SPEEDM[T1,T2,DIST]

calls: ABS, SPEEDAUX, FQUOTIENT
called by: SPEED, SPEEDFROM, SWR

STAMMER[]

calls: MAPC, WELCOME, EXLOOP, APPLYRULE, PARTING
uses free: PRODUCTIONS

STARTUP[]

calls: RETRIEVES, INCLUDEPLAT, MAPC, STUFFLN
called by: CKCONFIGURATION

STATE[L]

calls: CASSERT

STRIPSTREAM[S]

called by: GETATT, GETATTB, GETSTRIP, RETRIEVES, UNLESSHACK

STUFFLN[MLN]

calls: DSPADDTRH, MAPC, GETATT, DSPADDINC
called by: STARTUP
binds: VER

SUBTEND[LAT1,LON1,LAT2,LON2]

calls: EQP, ABS, FDIFFERENCE, COS, FPLUS, ARCCOS, FQUOTIENT, FTIMES, FMINUS
called by: DIRECTION, DISTANCE, LANERANGE
binds: C1,C2,C3

SUCCESSOR[SITE]

calls: LESSP, MAPC, GETATTB, GETATT, RETRIEVES
 called by: COURSE, SPEED
 binds: PLAT, TOSSITE, TOSX, SUCC, TOSSUC, X

SWEEPER[CONDITIONS, ACTIONS, EV]

calls: MEMB, SWEEPER, ORHACK, NOTHACK, UNLESSHACK, ANDHACK, CONSTRUCT,
 GETMRVAL, APPLY
 called by: SWEEPER, APPLYRULE, ORHACK, ANDHACK, NOTHACK, UNLESSHACK
 binds: THISCOND, C
 uses free: VDRELS

SWR[LT1, T1, LT2, T2]

calls: CENTROID, LESSP, SPEEDM, DISTANCE
 binds: L1, L2
 uses free: MAXSHIPSPEED

TEKCOM[STR]

calls: PRIN1, TERPRI
 called by: GRATEK, MONTEK
 uses free: TEKCOMCHAR

TEKTEST[]

calls: CLEARBUF, PRINCHAR, TERPRI, DISMISS, PRIN1, READP, ASKUSER, READ
 called by: CKCONFIGURATION
 binds: UTEKFLG
 uses free: TEKCOMCHAR, TEK4025, TEKFLG

TEKWAIT[]

calls: JSYS
 called by: WAITER, DSPERASE
 uses free: DSPTTYCODE, TEKFLG

TRACKINPOLY[TRACK, POLY]

calls: SOMELINESEG, FUNCTION, LINPOLY
 binds: TRACKPT1, TRACKPT2

TWO-PLACE[X]

calls: FQUOTIENT, FIX, FPLUS, FTIMES
 called by: MIDP, MODIFIER

UNCRUNCH[NUM]

calls: RPTQ, RPLACA, SETN, LRSH, LLSH
 called by: DSPNUMAT
 binds: PTR, RPTN
 uses free: SCRATCHFIVE

UNFREEZE[]

calls: MAPC,DREVERSE,APPLY*
 called by: MSGMTR
 binds: XSUSP
 uses free: FREEZELST,FREEZEFLG

UNLESSHACK[CONDITIONS,ACTIONS,EV]

calls: ORACLEHACK,CASSERT,STRIPSTREAM,RETSTREAM,MESSAGE1,MAPRETRIEVE,
 FUNCTION,LEQ,GETCON,SWEeper,CONS,LIST
 called by: SWEEPER
 binds: P,X,CLIST

VAR?[Q]

calls: CHCON1
 called by: MESSAGE1,ORACLEHACK,PREPALIST,RETSTREAM,RETvars,RETRIEVER

WAITER[]

calls: TERPRI,TEKWAIT,ASKUSER
 called by: DESCRIBMSG,PRINTRULEASSR
 uses free: DUALFLG

WEATHERMSG[txt]

calls: CASSERT,DSPADDTRH,MAPC,LIST,DSPADDINC,FLOAT
 called by: MSGMTR
 binds: SNAME,LOC,STVER,TM
 uses free: DISPLAYFLG

WELCOME[]

calls: PRIN1,TERPRI,MAPC,CKCONFIGURATION,LINEREAD,PUTPROP
 called by: STAMMER
 binds: NEWFL,TB
 uses free: ASSERTIONS,MSGFILE

WENT-AFTER[S1,T1,S2,T2,S3,T3,S4,T4]

calls: GREATERP,LESSP,CENTROID,DIRECTION,QUOTIENT,TIMES,COS,SIN,
 DIFFERENCE,LIST,PLUS,ARCTAN,ABS,DISTANCE
 binds: PHI,VM1,VM2,VP1,VP2,P0,P4,PSI,THETA,INITDIST,ENDDIST,MINDIST,
 MINTIME,L1,L2,L3,L4
 uses free: PATROLRANGE,MAXSHIPSPEED

WENT-BEFORE[S1,T1,S2,T2,S3,T3,S4,T4]

calls: LESSP,GREATERP,CENTROID,DIRECTION,QUOTIENT,TIMES,COS,SIN,
 DIFFERENCE,LIST,PLUS,ARCTAN,ABS,DISTANCE
 binds: PHI,VM1,VM2,VP1,VP2,P0,P4,PSI,THETA,INITDIST,FINDIST,MINDIST,
 MINTIME,L1,L2,L3,L4
 uses free: PATROLRANGE,MAXSHIPSPEED

WHAT2FORMFN[PL]

calls: APPEND, RETRIEVER, LIST

WHATFORMFN[REL, OBJ]

calls: RETRIEVER, LIST, MAPCAR, CDADR

binds: ANS

uses free: WHATRES, WHATRES2

WHOSE2FORMFN[VAL, REL]

calls: RETRIEVER, LIST, MAPCAR, CDADR

binds: ANS

uses free: WHOSE2RES, WHOSE2RES2

WITHINR[L]

calls: NCONC, MAKEFILE

uses free: WITHINRFNS

YESNO[ASSRSPEC]

calls: EQP, TERPRI, PRIN1, GAMF, GETSTRIP, GETCON

binds: NDECON, NDE

INITIAL DISTRIBUTION

NAVAL ELECTRONIC SYSTEMS COMMAND
CODE 330 (CC STOUT)
CODE PME-108 (D SCHUTZER)

(2)

ROME AIR DEVELOPMENT CENTER
GRIFFIS AFB, NY 13441
ISIS (N FOWLER III)

NAVAL RESEARCH LABORATORY
CODE 7509 (JH KULBACK)
CODE 7932 (HL WIENER)

NAVAL AIR DEVELOPMENT CENTER
CODE 6021 (LT S HARRIS)

(2)

DEFENSE ADVANCED RESEARCH PROJECTS
AGENCY

IPTO (DR R ENGELMORE)
IPTO (LCDR AJ DIETZLER)
IPTO (DR R KAHN)

(2)

(2)

MITRE CORP
PO BOX 208
BEDFORD, MA 01730
CARL ENGLEMAN

SYSTEMS DEVELOPMENT CORP
3065 ROSECRANS PLACE
SAN DIEGO, CA 92110
RJ BECHTEL
PH MORRIS
DF KIBLER

(4)

(4)

(4)

ESL INC
495 JAVA DR
SUNNYVALE, CA 94086
GK KIRENUDJIAN

SYSTEMS CONTROL INC
1801 PAGE MILL RD
PALO ALTO, CA 94304
JR PAYNE
RP WISHNER

MITRE CORP
WESTGATE RESEARCH PARK
MC LEAN, VA 22101
JW BENOIT

SYSTEMS EXPLORATION INC
1340 MUNRAS AVE
MONTEREY, CA 93940
R BUTTERWORTH
G GIBBONS

RAND CORP
1700 MAIN STREET
SANTA MONICA, CA 90406
G MARTINS
J GILLOGLY

SRI INTERNATIONAL
333 RAVENSWOOD AVE
MENLOW PARK, CA 94025
E SACERDOTI
J OLMSTEAD
D SAGALOWICZ

BOLT, BERANEK AND NEWMAN
50 MOULTON STREET
CAMBRIDGE, MA 02138
RJ BOBROW
NR GREENFIELD
O SELFRIDGE
J VITTAL

CTEC INC
7777 LEESBURG PIKE
FALLS CHURCH, VA 22043
KD SHERE

COMPUTER CORPORATION OF AMERICA
575 TECHNOLOGY SQ
CAMBRIDGE, MA 02139
GA WILSON
JB ROTHNIE

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93940
CODE 55 PK (G POOCK)
CODE 52 RL (RJ ROLAND)

DEFENSE TECHNICAL INFORMATION CENTER (12)

VERAC INC
4901 MORENA BLVD
SAN DIEGO, CA 92117
C MOREFIELD
J NASH
J TIERNAN

