

NOTICE

THIS DOCUMENT HAS BEEN REPRODUCED FROM
MICROFICHE. ALTHOUGH IT IS RECOGNIZED THAT
CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED
IN THE INTEREST OF MAKING AVAILABLE AS MUCH
INFORMATION AS POSSIBLE

NASA

National Aeronautics and
Space Administration

DEC 13 1980

JSC-17024

Lyndon B. Johnson Space Center
Houston, Texas 77058

Remiseo-15614

DETECTION AND MAPPING (DAM) PACKAGE

Volume 4a: Software System Manual (part 1)

NASA CR-161013

Edward H. Schlosser
Lockheed Engineering & Management Services Co., Inc.
1830 NASA Road 1
Houston, Texas 77058

NAS 9-15800

September 1980
Final Report for Period January - September 1980

(NASA-CR-161013) DETECTION AND MAPPING
(DAM) PACKAGE. VOLUME 4A: SOFTWARE SYSTEM
MANUAL, PART 1 Final Report, Jan. - Sep.
1980 (Lockheed Engineering and Management)
668 p HC A99/MF A01

N81-26749

Unclas
30068

CSSL 09B G3/61

Prepared for
LYNDON B. JOHNSON SPACE CENTER
Houston, Texas 77058



1. Report No.		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle DETECTION AND MAPPING (DAM) PACKAGE Volume 4a: Software System Manual (part 1)				5. Report Date September 1980	
				6. Performing Organization Code	
7. Author(s) Edward H. Schlosser				8. Performing Organization Report No. LEMSCO-15615	
9. Performing Organization Name and Address Lockheed Engineering & Management Services Co., Inc. 1830 NASA Road 1 Houston, Texas 77058				10. Work Unit No.	
				11. Contract or Grant No. NAS 9-15800	
12. Sponsoring Agency Name and Address O.G. Smith Earth Observation Division Lyndon B. Johnson Space Center, Houston, Texas 77058				13. Type of Report and Period Covered Final, Jan.-Sept. 1980	
				14. Sponsoring Agency Code	
15. Supplementary Notes Software available from: COSMIC University of Georgia 112 Barrow Hall Athens, Georgia 30602					
16. Abstract The DAM package is an integrated set of manual procedures, computer programs, and graphic devices designed for efficient production of precisely registered and formatted maps from digital Landsat multi-spectral scanner (MSS) data. The software can be readily implemented on any Univac 1100 series computer with standard peripheral equipment. This version of the software includes pre-defined spectral limits for use in classifying and mapping surface water for Landsat-1, Landsat-2, and Landsat-3. Tape formats supported include "X", "AM", and "PM".					
17. Key Words (Suggested by Author(s)) Computer programs, Earth resources, Earth satellites, land use, Landsat satellites, mapping, remote sensors, surface water, thematic mapping				18. Distribution Statement	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages	22. Price*

*For sale by the National Technical Information Service, Springfield, Virginia 22161

PREFACE

Multispectral scanners onboard NASA unmanned Landsat satellites provide an ideal source of current data for Earth resources applications. The Detection and Mapping (DAM) package was originally developed at the Johnson Space Center for rapid conversion of the Landsat digital data into hydrographic maps matching standard topographic quadrangle series. Recent improvements in both the manual procedures and computer programs within the DAM package make it easier to use, faster, and more general purpose.

Documentation and software for the DAM package are available to all public and private agencies, in accordance with the NASA policy of encouraging maximum use of remote sensing technology.

Published documentation, in which this is volume 4a, is comprised of the following volumes:

Volume 1: General Procedure

Volume 2: Software User Manual (in two parts)

Volume 3: Control Network Establishment

Volume 4: Software System Manual (in two parts)

These volumes supersede the previous documentation published in 1973. Software releases prior to version 7602 cannot be used with the current documentation.

Volume 4a contains software listings and documentation which have not been published prior to version 8009.

PRECEDING PAGE BLANK NOT FILMED

I 0009

DETECTION AND MAPPING PACKAGE
.....

SYSTEM DESIGN

E H SCHLOSSER

PROGRAMMING

D A BECK
M L BROWN
J C CRISP
H O EPPLER
C A HELMKE
W A HOLLEY
T R KELL
R E HARVESON
J C POOLEY
E H SCHLOSSER
M A TOMPKINS

FUNDING

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B JOHNSON SPACE CENTER
HOUSTON, TEXAS 77058

CONTRACTOR

LOCKHEED ENGINEERING AND MANAGEMENT SERVICES COMPANY
HOUSTON, TEXAS 77058

PRECEDING PAGE BLANK NOT FILMED

8ASG.CP LISTPRINT..F/32/TRK/512
 8BRKPT PRINTS/LISTPRINT
 8HDD.P XXX DAM PACKAGE APPENDICES (VERSION 8009)
 8PRT.F DAM.
 8PRT.S DAM.SYS-TITLE
 8PRT.S DAM.SYS-LIST
 8HDD.P XXX DAM PACKAGE APPENDIX A: GENERAL DOCUMENTATION
 8ADD DAM.APPENDIX-A
 8HDD.P XXX DAM PACKAGE APPENDIX B: EXEC COMMAND DOCUMENTATION
 8ADD DAM.APPENDIX-B
 8HDD.P XXX DAM PACKAGE APPENDIX C: PROGRAM USER DOCUMENTATION
 8ADD DAM.APPENDIX-C
 8HDD.P XXX DAM PACKAGE APPENDIX D: COMMAND USER DOCUMENTATION
 8ADD DAM.APPENDIX-D
 8HDD.P XXX DAM PACKAGE APPENDIX E: MACRO COMMAND DOCUMENTATION
 8ADD DAM.APPENDIX-E
 8HDD.P XXX DAM PACKAGE APPENDIX F: SAMPLE CONTROL NETWORKS
 8ADD DAM.APPENDIX-F
 8HDD.P XXX DAM PACKAGE APPENDIX G: SPECTRAL LIMITS/TRANSFORMS
 8ADD DAM.APPENDIX-G
 8HDD.P XXX DAM PACKAGE APPENDIX H: SAMPLE RUNSTREAMS
 8ADD DAM.APPENDIX-H
 8HDD.P XXX DAM PACKAGE APPENDIX I: REVISIONS AND NEWS
 8ADD DAM.APPENDIX-I
 8HDD.P XXX DAM PACKAGE APPENDIX J: DEFAULT COMMANDS
 8ADD DAM.APPENDIX-J
 8HDD.P XXX DAM PACKAGE APPENDIX K: SYSTEM IMPLEMENTATION
 8ADD DAM.APPENDIX-K
 8HDD.P XXX DAM PACKAGE APPENDIX L: MAIN PROGRAMS/ROUTINES
 8ADD DAM.APPENDIX-L
 8HDD.P XXX DAM PACKAGE APPENDIX M: COMMAND ROUTINES
 8ADD DAM.APPENDIX-M
 8HDD.P XXX DAM PACKAGE APPENDIX N: UTILITY ROUTINES
 8ADD DAM.APPENDIX-N
 8HDD.P XXX DAM PACKAGE APPENDIX O: COORDINATE TRANSFORMATIONS
 8ADD DAM.APPENDIX-O
 8HDD.P XXX DAM PACKAGE APPENDIX P: EXECUTIVE REQUESTS
 8ADD DAM.APPENDIX-P
 8HDD.P XXX DAM PACKAGE APPENDIX Q: MACROS
 8ADD DAM.APPENDIX-Q
 8HDD.P XXX DAM PACKAGE APPENDIX R: CHAR/BYTE/STRING ROUTINES
 8ADD DAM.APPENDIX-R
 8HDD.P XXX DAM PACKAGE APPENDIX S: SORT ROUTINES
 8ADD DAM.APPENDIX-S
 8HDD.P XXX DAM PACKAGE APPENDIX Y: INTERNAL CODE/TESTING
 8ADD DAM.APPENDIX-Y
 8HDD.P XXX DAM PACKAGE APPENDIX Z: FILE DIRECTORY
 8PRT.TL DAM.
 8BRKPT PRINTS
 8FREE LISTPRINT.
 8PRT.F LISTPRINT.
 8SYM LISTPRINT.

PRECEDING PAGE BLANK NOT FILMED

(This volume contains Appendices I thru M)

DAM PACKAGE APPENDIX A
GENERAL DOCUMENTATION

APPENDIX-A
001

SPRT.SC DAM.	APPENDIX-A . (0009)..A-1
SPRT.SC DAM.EXP----	.
SPRT.SC DAM.EXP-DAM	. DAM PACKAGE.....A-3
SPRT.SC DAM.EXP----
SPRT.SC DAM.EXP-LANDSAT	. LANDSAT.....A-5
SPRT.SC DAM.EXP-RULES	. RULES.....A-6.
SHORE:	. A-7
SPRT.SC DAM.EXP-FILES	. FILES.....A-8
SPRT.SC DAM.EXP-CARD	. CARD CODES.....A-9
SPRT.SC DAM.EXP-COORDINA	. COORDINATES.....A-10
SPRT.SC DAM.EXP-COMPUTER	. COMPUTER RUNS.....A-11
SPRT.SC DAM.EXP-LOCAL	. LOCAL STANDARDS.....A-12

DAN PACKAGE APPENDIX B
EXEC COMMAND DOCUMENTATION

APPENDIX-B
001

SPRT.SC DAN.	APPENDIX-B . (0009)..B-1
SPRT.SC DAN.EXP----	.
SPRT.SC DAN.EXP-EXEC	. SUMMARY.....B-3
SPRT.SC DAN.EXP-SADD	. SADD.....B-4
SPRT.SC DAN.EXP-SASO	. SASO.....B-5
SPRT.SC DAN.EXP-SCOPY	. SCOPY.....B-6
SPRT.SC DAN.EXP-SDATA/CHECKOUT	. SDATA/CHECKOUT.....B-7
SPRT.SC DAN.EXP-SED	. SED.....B-8
SPRT.SC DAN.EXP-SEND	. SEND.....B-9
SPRT.SC DAN.EXP-SEOF	. SEOF.....B-10
SPRT.SC DAN.EXP-SFIN	. SFIN.....B-11
SPRT.SC DAN.EXP-SFREE	. SFREE.....B-12
SPRT.SC DAN.EXP----
SPRT.SC DAN.EXP-SLOCATE	. SLOCATE.....B-14
SPRT.SC DAN.EXP----
SPRT.SC DAN.EXP----
SPRT.SC DAN.EXP-SREWIND	. SREWIND.....B-17
SPRT.SC DAN.EXP-SRUN	. SRUN.....B-18
SPRT.SC DAN.EXP-SUSE	. SUSE.....B-19
SPRT.SC DAN.EXP-SUSHAP	. SUSHAP.....B-20
SPRT.SC DAN.EXP-SXQT	. SXQT.....B-21
SPRT.SC DAN.EXP----	.

DAM PACKAGE APPENDIX C
PROGRAM USER DOCUMENTATION

APPENDIX-C
001

SPRT.SC DAM.	APPENDIX-C . (0005)..C-1
SPRT.SC DAM.EXP----	.
SPRT.SC DAM.EXP-PROGRAMS	. SUMMARY.....C-3
SPRT.SC DAM.EXP-ERTS-DUP	. ERTS-DUP.....C-4
SPRT.SC DAM.EXP-ERTSIDC	. ERTSIDC.....C-5
SPRT.SC DAM.EXP-PICTAB	. PICTAB.....C-6.
SHORE:	. C-7
SPRT.SC DAM.EXP-CONTROL	. CONTROL.....C-8.
SHORE:	. C-9
SPRT.SC DAM.EXP-CLASSIFY	. CLASSIFY.....C-10.
SHORE:	. C-11
SPRT.SC DAM.EXP-PRTDET	. PRTDET.....C-12.
SHORE:	. C-13
SPRT.SC DAM.EXP-PRTCLASS	. PRTCLASS.....C-14.
SHORE:	. C-15
SPRT.SC DAM.EXP-PLTCLASS	. PLTCLASS.....C-16
SPRT.SC DAM.EXP----
SPRT.SC DAM.EXP-STATUS	. STATUS.....C-18

DAN PACKAGE APPENDIX D
 COMMAND USER DOCUMENTATION

APPENDIX-D
 001

SPRT.SC DAN.	APPENDIX-D . (0000)..D-1.	SET TABS 8 27 & 30
SHORE:	. D-2	
SPRT.SC DAN.EXP-COMMANDS	. SUMMARY.....D-3.	
SHORE:	. D-4.	
SHORE:	. D-5	(FUTURE)
SPRT.SC DAN.EXP----	
SPRT.SC DAN.EXP-ADJ	. ADJUST.....D-7	
SPRT.SC DAN.EXP-ALI	. ALIGN.....D-8	
SPRT.SC DAN.EXP----	ALTITUDE (FUTURE)
SPRT.SC DAN.EXP-ATT	. ATTITUDE.....D-10	
SPRT.SC DAN.EXP-CEN	. CENTER.....D-11	
SPRT.SC DAN.EXP-CHA	. CHANNEL.....D-12	
SPRT.SC DAN.EXP-CLE	. CLEAR.....D-13	
SPRT.SC DAN.EXP-COL	. COLOR.....D-14	
SPRT.SC DAN.EXP-COP	. COPIES.....D-15	
SPRT.SC DAN.EXP-CON	. COUNT.....D-16	
SPRT.SC DAN.EXP-CRO	. CROSSTAB.....D-17	
SPRT.SC DAN.EXP----	CRT (FUTURE)
SPRT.SC DAN.EXP-DEN	. DENSITY.....D-19	
SPRT.SC DAN.EXP-DET	. DETECT.....D-20	
SPRT.SC DAN.EXP-DIA	. DIAGRAM.....D-21	
SPRT.SC DAN.EXP-DIS	. DISPLAY.....D-22	
SPRT.SC DAN.EXP-EXI	. EXIT.....D-23	
SPRT.SC DAN.EXP-EXP	. EXPLAIN.....D-24	
SPRT.SC DAN.EXP-FAC	. FACTOR.....D-25	
SPRT.SC DAN.EXP-FI	. FI.. (ENDIF).....D-26	
SPRT.SC DAN.EXP-GEO	. GEOMETRY.....D-27	
SPRT.SC DAN.EXP----	(FUTURE)
SPRT.SC DAN.EXP-HEA	. HEADING.....D-29	
SPRT.SC DAN.EXP----	. HISTOGRAM.....D-30	HISTOGRAM (FUTURE)
SPRT.SC DAN.EXP-IF	. IF.....D-31	
SPRT.SC DAN.EXP-INT	. INTENSITY.....D-32	
SPRT.SC DAN.EXP-LIN	. LINEAR.....D-33	
SPRT.SC DAN.EXP-LIS	. LIST.....D-34	
SPRT.SC DAN.EXP-MAP	. MAP.....D-35	
SPRT.SC DAN.EXP-MER	. MERIDIAN.....D-36	
SPRT.SC DAN.EXP----	MODEL (FUTURE)
SPRT.SC DAN.EXP-NAM	. NAME.....D-38	
SPRT.SC DAN.EXP-NEW	. NEWS.....D-39	
SPRT.SC DAN.EXP-NEX	. NEXT.....D-40	(REPLACE WITH IF...FI)
SPRT.SC DAN.EXP-OFF	. OFF.....D-41	
SPRT.SC DAN.EXP-ON	. ON.....D-42	
SPRT.SC DAN.EXP-ORI	. ORIGIN.....D-43	
SPRT.SC DAN.EXP----	(FUTURE)
SPRT.SC DAN.EXP-PAO	. PAGE.....D-45	
SPRT.SC DAN.EXP----	PARTITION (FUTURE)
SPRT.SC DAN.EXP-PEE	. PEEK.....D-47	
SPRT.SC DAN.EXP-PIC	. PICTURE.....D-48	
SPRT.SC DAN.EXP----	. PLOTTER.....D-49	
SPRT.SC DAN.EXP-POI	. POINT.....D-50	
SPRT.SC DAN.EXP-POK	. POKE.....D-51	
SPRT.SC DAN.EXP-POL	. POLAR.....D-52	
SPRT.SC DAN.EXP-PRI	. PRINTER.....D-53	
SPRT.SC DAN.EXP-PRO	. PROFILE.....D-54	
SPRT.SC DAN.EXP----	(FUTURE)
SPRT.SC DAN.EXP-RAD	. RADIANCE.....D-56	

DAN PACKAGE APPENDIX D
COMMAND USER DOCUMENTATION

APPENDIX-D
002

SPRT.SC DAN.EXP-RAN . RANK.....D-57
SPRT.SC DAN.EXP-REN . RENUNBER.....D-58
SPRT.SC DAN.EXP-RES . RESAMPLING.....D-59
SPRT.SC DAN.EXP-ROT . ROTATE.....D-60
SPRT.SC DAN.EXP----D-61
SPRT.SC DAN.EXP-SCA . SCALE.....D-62
SPRT.SC DAN.EXP-SCE . SCENE.....D-63
SPRT.SC DAN.EXP----D-64
SPRT.SC DAN.EXP-SHA . SHARPENING.....D-65
SPRT.SC DAN.EXP-SIZ . SIZE.....D-66
SPRT.SC DAN.EXP----D-67
SPRT.SC DAN.EXP-SPA . SPACING.....D-68
SPRT.SC DAN.EXP---- . SPHEROID.....D-69
SPRT.SC DAN.EXP-SYM . SYMBOLS.....D-70
SPRT.SC DAN.EXP-TAB . TABULATE.....D-71
SPRT.SC DAN.EXP----
SPRT.SC DAN.EXP-TIC . TICK.....D-73
SPRT.SC DAN.EXP-TIM . TIME.....D-74
SPRT.SC DAN.EXP-TIN . TINT.....D-75
SPRT.SC DAN.EXP-TOL . TOLERANCE.....D-76
SPRT.SC DAN.EXP-TOT . TOTAL.....D-77
SPRT.SC DAN.EXP----
SPRT.SC DAN.EXP-WIN . WINDOW.....D-79
SPRT.SC DAN.EXP-ZON . ZONE.....D-80

SATURATION (FUTURE)

SCRIPT (FUTURE)

SKEW (FUTURE)

SPHEROID (FUTURE)

TERNINAL (FUTURE)

(FUTURE)

DAN PACKAGE APPENDIX E
 MACRO COMMAND DOCUMENTATION

APPENDIX-E
 001

SET TAGS 8 20 6 30

SPRT.SC	DAN.	APPENDIX-E . (0000).....	E-1
SPRT.SC	DAN.EXP----	
SPRT.SC	DAN.EXP-MACRO	USING MACRO COMMANDS.....	E-3
SPRT.SC	DAN.EXP-MACDEF	DEFINING MACRO COMMANDS...	E-4
SPRT.SC	DAN.PIC-ANO-RAD(PICTAB).....	E-5
SPRT.SC	DAN.CLA-CIR-20(CLASSIFY)....	E-6
SPRT.SC	DAN.PIC-CIR-20(PICTAB).....	E-7
SPRT:			E-8
SPRT.SC	DAN.PRO-CIR-20(PRTOET).....	E-9
SPRT.SC	DAN.CLA-CONFIRM(CLASSIFY)....	E-10
SPRT.SC	DAN.CON-CONFIRM(CONTROL)....	E-11
SPRT.SC	DAN.PIC-CONFIRM(PICTAB).....	E-12
SPRT.SC	DAN.PRC-CONFIRM(PRCLASS)....	E-13
SPRT.SC	DAN.CLA-HELP(CLASSIFY).....	E-14
SPRT.SC	DAN.CON-HELP(CONTROL).....	E-15
SPRT.SC	DAN.PIC-HELP(PICTAB).....	E-16
SPRT.SC	DAN.PRC-HELP(PRCLASS).....	E-17
SPRT.SC	DAN.EXP----	
SPRT.SC	DAN.	JSC-PRINT	E-18
SPRT.SC	DAN.EXP----	
SPRT.SC	DAN.PIC-LIST-ALL(PICTAB).....	E-21
SPRT.SC	DAN.PIC-LIST-9(PICTAB).....	E-22
SPRT.SC	DAN.	HAC4SPEC-1-2	E-23
SPRT.SC	DAN.EXP----	
SPRT.SC	DAN.	NED-PRINT	E-25
SPRT.SC	DAN.EXP----	
SPRT.SC	DAN.PRC-QISNAP	E-27
SPRT.SC	DAN.PIC-QISSPEC	E-28
SPRT.SC	DAN.	SED-PRINT	E-29
SPRT.SC	DAN.EXP----	
SPRT.SC	DAN.PIC-STAR-KM	E-31
SPRT.SC	DAN.	STAR-SCAN	E-32
SPRT.SC	DAN.	WES-PRINT	E-33
SPRT.SC	DAN.EXP----	
SPRT.SC	DAN.	50X50H-L0	E-35
SPRT.SC	DAN.	50X50H-L0-R2	E-36
SPRT.SC	DAN.	50X50H-L0-R3	E-37
SPRT.SC	DAN.	50X50H-L0-R4	E-38
SPRT.SC	DAN.	50X50H-L0-R5	E-39
SPRT.SC	DAN.EXP----	
SPRT.SC	DAN.	50X50H-L0	E-41
SPRT.SC	DAN.	50X50H-L0-R2	E-42
SPRT.SC	DAN.	50X50H-L0-R3	E-43
SPRT.SC	DAN.	50X50H-L0-R4	E-44

ORIGINAL PAGE IS
 OF POOR QUALITY

DAN PACKAGE APPENDIX F
SAMPLE CONTROL NETWORKS

APPENDIX-F
001

SPRT.SC DAN.APPENDIX-F . (0000).....F-1	
SPRT.SC	DAN.EXP----
SPRT.SC DAN.1037-10244F-3	
SPRT.SC DAN.1037-10244/UTMF-4	
SPRT.SC DAN.1073-10244-3 ..(STRIP 3)....F-5	
SPRT.SC DAN.1092-10308F-6	
SPRT.SC DAN.1092-10308-3 ..(STRIP 3)....F-7	
SPRT.SC	DAN.EXP----
SPRT.SC DAN.1132-10812F-8	
SPRT.SC DAN.1191-10301F-10	
SPRT.SC DAN.1205-10404F-11	
SPRT.SC DAN.1205-10010F-12	
SPRT.SC DAN.1209-10201F-13	
SPRT.SC DAN.1302-10551F-14	
SPRT.SC DAN.1407-10301F-15	
SPRT.SC DAN.1420-10303F-16	
SPRT.SC DAN.1420-10305F-17	
SPRT.SC DAN.1704-10231F-18	
SPRT.SC DAN.1700-20351F-19	
SPRT.SC	DAN.EXP----
SPRT.SC DAN.21054-17513 .. ('PH')F-21	
SPRT.SC	DAN.EXP----
SPRT.SC DAN.30130-10032 .. ('X')F-23	
SPRT.SC	DAN.EXP----

DAN PACKAGE APPENDIX 6
SPECTRAL LIMITS/TRANSFORMS

APPENDIX-6
881

SPRT.SC DAN.APPENDIX-6 . (0000)..0-1.
 SHORE: . 0-2
 SPRT.SC DAN.WATER-LIN 0-3
 SPRT.SC DAN.WATER-LIN/1 0-4
 SPRT.SC DAN.WATER-LIN/1-ANDERSON 0-5
 SPRT.SC DAN.WATER-LIN/1-4IN-NARROW 0-6
 SPRT.SC DAN.WATER-LIN/1-14N-NARROW 0-7
 SPRT.SC DAN.WATER-LIN/1-4IN-TURBNI 0-8
 SPRT.SC DAN.WATER-LIN/1-14N-TURBNI 0-9
 SPRT.SC
 SPRT.SC DAN.WATER-LIN/2 0-11
 SPRT.SC DAN.WATER-LIN/2A 0-12
 SPRT.SC DAN.WATER-LIN/2A-124F-PURE 0-13
 SPRT.SC DAN.WATER-LIN/2A-124H-PURE 0-14
 SPRT.SC DAN.WATER-LIN/2A-124N-PURE 0-15
 SPRT.SC DAN.WATER-LIN/2A-14F-PURE 0-16
 SPRT.SC DAN.WATER-LIN/2A-14H-PURE 0-17
 SPRT.SC DAN.WATER-LIN/2A-14N-PURE 0-18
 SPRT.SC DAN.WATER-LIN/2A-412F-PURE 0-19
 SPRT.SC DAN.WATER-LIN/2A-14F-PURE 0-20
 SPRT.SC DAN.WATER-LIN/2A-412F-PURE 0-21
 SPRT.SC DAN.WATER-LIN/2A-14F-PURE 0-22
 SPRT.SC
 SPRT.SC DAN.CHAN-CALIB/2B 0-24
 SPRT.SC DAN.WATER-LIN/2B 0-25
 SPRT.SC DAN.WATER-LIN/2B-14N-PURE 0-26
 SPRT.SC DAN.WATER-LIN/2B-124N-NIXI 0-27
 SPRT.SC DAN.WATER-LIN/2B-124F-PURE 0-28
 SPRT.SC DAN.WATER-LIN/2B-124H-PURE 0-29
 SPRT.SC DAN.WATER-LIN/2B-124N-PURE 0-30
 SPRT.SC DAN.WATER-LIN/2B-14F-PURE 0-31
 SPRT.SC DAN.WATER-LIN/2B-14H-PURE 0-32
 SPRT.SC DAN.WATER-LIN/2B-14N-PURE 0-33
 SPRT.SC DAN.WATER-LIN/2B-412F-PURE 0-34
 SPRT.SC DAN.WATER-LIN/2B-14F-PURE 0-35
 SPRT.SC DAN.WATER-LIN/2B-412F-PURE 0-36
 SPRT.SC DAN.WATER-LIN/2B-14F-PURE 0-37
 SPRT.SC
 SPRT.SC DAN.CHAN-CALIB/2D 0-39
 SPRT.SC DAN.WATER-LIN/2D 0-40
 SPRT.SC DAN.WATER-LIN/2D-412F-PURE 0-41
 SPRT.SC
 SPRT.SC DAN.WATER-LIN/3 0-43
 SPRT.SC DAN.WATER-LIN/3A 0-44
 SPRT.SC DAN.WATER-LIN/3A-LIN-NI 0-45
 SPRT.SC
 SPRT.SC DAN.CHAN-CALIB/3C 0-47
 SPRT.SC DAN.WATER-LIN/3C 0-48
 SPRT.SC DAN.WATER-LIN/3C-LIN-NI 0-49
 SPRT.SC
 SPRT.SC DAN.WATER-NESS/1 0-51
 SPRT.SC DAN.WATER-NESS/2A 0-52
 SPRT.SC DAN.WATER-NESS/2B 0-53
 SPRT.SC DAN.WATER-NESS/2D 0-54
 SPRT.SC DAN.WATER-NESS/3A 0-55
 SPRT.SC DAN.WATER-NESS/3C 0-56

DAN.EXP----

DAN.EXP----

DAN.EXP----

DAN.EXP----

DAN.EXP----

DAN.EXP----

**DAN PACKAGE APPENDIX 0
SPECTRAL LIMITS/TRANSFORMS**

**APPENDIX-0
002**

**SPRT.SC
SPRT.SC DAN.TURBID-NESS/10-58
SPRT.SC DAN.TURBID-NESS/2A0-59
SPRT.SC DAN.TURBID-NESS/2B0-60**

DAN.EXP----

DAN PACKAGE APPENDIX H
SAMPLE RUNSTREAMS

APPENDIX-H
001

SPRT.SC DAN.	APPENDIX-H . (0000).....H-1
SPRT.SC DAN.EXP----	.
SPRT.SC DAN.EXP----	.
SPRT.SC DAN.RUN-ERTS-DUP	. ERTS-DUP.....H-4
SPRT.SC DAN.RUN-ERTS-DUP/JSC	. ERTS-DUP(JSC).....H-5
SPRT.SC DAN.EXP----	.
SPRT.SC DAN.RUN-PICTAB/X-SF	. PICTAB(SINGLE FILE 'X' TAPES)..H-7
SPRT.SC DAN.RUN-PICTAB/X-MF	. PICTAB(MULTI-FILE 'X' TAPES)...H-8
SPRT.SC DAN.RUN-PICTAB/PH	. PICTAB('PH' TAPES).....H-9
SPRT.SC DAN.EXP----	.
SPRT.SC DAN.RUN-CONTROL	. CONTROL.....H-11
SPRT.SC DAN.EXP----	.
SPRT.SC DAN.RUN-CLA-PRC/X	. CLASSIFY/PRTCLASS('X' TAPES)...H-13
SPRT.SC DAN.RUN-CLA-PRC/PH	. CLASSIFY/PRTCLASS('PH' TAPES)..H-14
SPRT.SC DAN.EXP----	.
SPRT.SC DAN.RUN-STATUS	. STATUS.....H-16

**DAN PACKAGE APPENDIX I
REVISIONS AND NEWS**

**APPENDIX-I
001**

**SPRT.SC DAN.APPENDIX-I
SPRT.SC DAN.REV-DAN
SPRT.SC DAN.NEW-DAN
SPRT.SC DAN.NEW-PLTCLASS**

**DAM PACKAGE APPENDIX I
REVISIONS AND NEWS**

**REV-DAM
001**

DAN PACKAGE APPENDIX I
REVISIONS AND NEWS

NEW-DAN
001

1
0
0NEWS.....NEWS.....DAN.....NEWS.....NEWS.....
07802: PLYCLASS NOT AVAILABLE (USE VERSION 9-06)
07801: ACCEPT JULY 77 TAPE ANNOTATION RECORD
07802: EXTRACT ATTITUDE & CALIBRATION DATE FROM SIAT FILE
7802: ACCEPT JAN 78 TAPE ID RECORD
7802: LANDSAT-2 WATER LIMITS & TRANSFORMS
07804: REVISED WATER LIMITS & TRANSFORMS
7804: FIXES FOR UNIVAC PRINTER I/O BUGS
07807: REVISED PRINTER & DENSITY COMMANDS
7807: NEW DITCOP PROGRAM
7807: NEW TOLERANCE COMMAND
7807: MORE OFF/ON MODE OPTIONS
7807: LARGER DISPLAY/MAP PRINT FILES
7807: FIX DETECTION FILE READ2N BUFFER LINE ERROR
07903: STATUS FINDS BACKLOG & ACTIVE RUNS
7903: CLASSIFY MARKS DETECTION FILE BAD AFTER FATAL TAPE ERROR
7903: MACRO COMMANDS
7903: DEFAULT/MACRO COMMANDS MAY BE IN PRIVATE USER FILE
7903: COUNT & TIME COMMANDS
7903: DETECT.RADIANCE
7903: REV WATER LIMITS & TRANSFORMS FOR VERY HIGH SUN ANGLE
07912: PX8DEF INTERNAL BUFFER STRUCTURE
7912: HANDLE LANDSAT-1/-2/-3 BIL 'AM' & 'PM' TAPES
7912: HANDLE LANDSAT-3 BIP 'X' TAPES
7912: NEGATIVE RADIANCE IN TRANSFORMED CHANNEL CONVERTED TO ZERO
7912: RANGE FOR RADIANCE DETECTION FILES EXPANDED TO 0-127
7912: NON-INTEGGER SPACING
7912: GEOMETRY, CENTER, SIZE, LIST COMMANDS
7912: IF, FI, PEEK, POKE, COLOR, INTENSITY, PICTURE COMMANDS
7912: LANDSAT-2D ('PM' TAPES) WATER LIMITS & TRANSFORMS
08005: MULTI-CHANNEL RADIANCE DETECTION FILES
8005: PRIDENS CHANGED TO PRIDET
8005: OPTIONAL DISK INPUT TO PICTAB & CLASSIFY
8005: COLOR GRAPHICS INTERFACE
08009: LANDSAT-3A & -3C WATER LIMITS & TRANSFORMS
8009: TAPE QHD STATUS FROM FITEMS OR INFOS
8009: FIX HSKPIX BIN TYPE ERROR WHEN OVER 2 VERTICES
0
0NEWS.....NEWS.....JSC.....NEWS.....NEWS.....
0JSC FIX #8: IN BATCH, INSERT %PRT.T BETWEEN %USE & %ASO, AS BELOW:
 %USE DAN..<NAME OF DAN FILE>
 %PRT.T
 %ASO.A DAN.
0

ORIGINAL PAGE IS
OF POOR QUALITY

DAN PACKAGE APPENDIX I
REVISIONS AND NEWS

NEW-PLTCLASS
001

1

PLTCLASS NEWS

VERSION 7002:

1. PLTCLASS NOT CURRENTLY SUPPORTED (USE VERSION 5-06)

DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

APPENDIX-J
001

SPRT.SC DAN.APPENDIX-J
SPRT.SC DAN.DEF-ERTSIOC
SPRT.SC DAN.DEF-PICTAB
SPRT.SC DAN.DEF-CONTROL
SPRT.SC DAN.DEF-CLASSIFY
SPRT.SC DAN.DEF-PRYDET
SPRT.SC DAN.PRO-DEF-RAD
SPRT.SC DAN.PRO-DEF-DEN
SPRT.SC DAN.PRO-DEF-CLA
SPRT.SC DAN.DEF-PRYCLASS
SPRT.SC DAN.PRC-DEF-RAD
SPRT.SC DAN.PRC-DEF-DEN
SPRT.SC DAN.PRC-DEF-CLA
SPRT.SC DAN.DEF-PLTCLASS
SPRT.SC DAN.DEF-STATUS

**DAN PACKAGE APPENDIX J
DEFAULT COMMANDS**

**DEF-ERTSIDC
001**

***THERE ARE NO DEFAULT COMMANDS FOR ERTSIDC**

DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

DEF-PICTAB
001

ON.DEFAULT..PICTAB DEFAULT COMMANDS

NAME. .. (NO NAME)
LINEAR.1.WEIGHTS.1.0.0.0 ... LINEAR.1.GAIN.1.BIAS.-0 .. (CHAN 1) - 0
LINEAR.2.WEIGHTS.0.0.0.1 ... LINEAR.2.GAIN.1.BIAS.0 .. (CHAN 4)
POLAR.1.GAIN.1.BIAS.0 ... POLAR.2.GAIN.57.298 DEG/RADIAN.BIAS.0
PRINTER.FILES.1
ON.PROMPT.CONFIRM.LEGEND
COPIES.1
SYN.0.0.9.9...SYN.A.10.J.19...SYN.K.20.T.39...SYN.U.40.Z.127
CHANNEL.4
RADIANCE.0.14
TICK.SCAN.3000.3500.SCAN.10.10.. NO PRIMARY TICKS -- SECONDARY 10 LINES & COL
ZONE.. (TO CONFIRM ZONE IF CURRENTLY DEFINED)
IF.OFF.BATCH!..MIN.PRINT.-0.-25.+0.+25..17LIN X 51COL...SPA.3.2...F1
IF.ON.BATCH...MIN.PRINT.-180.-60.+180.+60..36IX12I...SPA.1.1...ON.ECHO...F1
OFF.DEFAULT..PICTAB

DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

DEF-CONTROL
001

ON.DEFAULT..CONTROL DEFAULT COMMANDS
ON.PROMPT.CONFIRM
OFF.DEFAULT..CONTROL

**DAN PACKAGE APPENDIX J
DEFAULT COMMANDS**

**DEF-CLASSIFY
001**

**ON.DEFAULT..CLASSIFY DEFAULT COMMANDS
TOLERANCE.0
ON.PROMPT.CONFIRM
COPIES.1
IF.OFF.BATCH...WINDOW.SCAN.-80.-400.+80.+400..161LIN X 801SAH...F!
IF.ON.BATCH...ORIG.SCAN.1.1...MIN.SCAN.2500.3500..ENTIRE SCENE...ON.ECHO...F!
OFF.DEFAULT..CLASSIFY**

DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

DEF-PRDET
001

ON.DEFAULT .. *** PRDET DEFAULT COMMANDS ***
PRINTER.FILES.1
ON.PROMPT.CONFIRM.LEGEND
COPIES.1
MACYSPEC-1-2.DEF.(DETECT) .. DEFAULTS FOR CURRENT TYPE OF DETECTION FILE
SPACING.1.1
ZONE
IF.OFF.BATCH ... WINDOW.PRINT. -8.-25. +8.+25.. 17 LINE X 51 COL ... F1
IF.ON.BATCH ... WINDOW.PRINT. -100.-60. +100.+60.. 36X121 ... ON.ECHO ... F1
OFF.DEFAULT..PRDET

DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

PRO-DEF-RAD
001

DEF-RAD .. PROCT MACRO COMMAND WITH DEFAULTS FOR RADIANCE DETECTION FILE
RADIANCE.0.127
SYN.0.0.0.0...SYN.A.10.J.10...SYN.K.20.T.30...SYN.U.40.Z.127

REPRODUCED FROM THE
DAN PACKAGE

DEF-DEF .. PRDEF MACRO COMMAND WITH DEFAULTS FOR DENSITY DETECTION FILE
 DENSITY.0.10
 SYN.0.0.0.0...SYN.A.10.J.10

• NUMBER OF • NEIGHBORING • PIXELS OF • NAMED CLASS	CENTRAL PIXEL CLASSIFICATION STATUS					
	OTHER CLASS		NAMED CLASS		NO DATA	
• -----	DEF	SYN	DEF	SYN	DEF	SYN
• 0	00	0	10	A	20	:
• 1	01	1	11	B	21	:
• 2	02	2	12	C	22	:
• 3	03	3	13	D	23	:
• 4	04	4	14	E	24	:
• 5	05	5	15	F	25	:
• 6	06	6	16	G	26	:
• 7	07	7	17	H	27	:
• 8	08	8	18	I	28	:
• 9	09	9	19	J	29	:

**DAN PACKAGE APPENDIX J
DEFAULT COMMANDS**

**PRO-DEF-CLA
001**

**DEF-CLA .. PRTOET MACRO COMMAND WITH DEFAULTS FOR CLASS DETECTION FILE
CLASS.0.61
SYN.0.0.9.9...SYN.A.10.J.19...SYN.K.20.T.39...SYN.U.40.Z.61**

DAH PACKAGE APPENDIX J
DEFAULT COMMANDS

DEF-PRCLASS
001

ON.DEFAULT..PRCLASS DEFAULT COMMANDS
ON.PROMPT.CONFIRM.LEGEND
COPIES.1
MACYSPEC-1-2.DEF.(DETECT) .. DEFAULTS FOR CURRENT TYPE OF DETECTION FILE
SCALE.1/24000
WINDOW.MINUTES.7.5.7.5
TICK INTERVAL.MINUTES.7.5.7.5.MINUTES.2.5.2.5
ZONE
IF.ON.BATCH ... ON.ECHO ... OFF.CONFIRM ... F1
OFF.DEFAULT..PRCLASS

DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

PRC-DEF-RAD
001

DEF-RAD .. PRTCLASS MACRO COMMAND WITH DEFAULTS FOR RADIANCE DETECTION FILE
RADIANCE.0.127
COUNT.0
SYN.0.0.9.9...SYN.A.10.J.19...SYN.K.20.T.39...SYN.U.40.Z.127

DAM PACKAGE APPENDIX J
DEFAULT COMMANDS

PRC-DEF-DEF
001

DEF-DEF .. PRTCLASS MACPO COMMAND WITH DEFAULTS FOR DENSITY DETECTION FILE
DENSITY.10.19
COUNT.1
SYN. .0...SYN.1.1.9.9...SYN.A.10.R.99...SYN.S.100.Y.169...SYN.Z.170.255

DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

PRC-DEF-CLA
001

DEF-CLA .. PRTCLASS MACRO COMMAND WITH DEFAULTS FOR CLASS DETECTION FILE
CLASS.0.61
COUNT.0
SYN.0.0.9.9...SYN.A.10.J.19...SYN.K.20.T.39...SYN.U.40.Z.61

DAM PACKAGE APPENDIX J
DEFAULT COMMANDS

DEF-PLTCLASS
001

ON.DEFAULT..PLTCLASS
ON.CONFIRM
0.99..59.
1.LINE
DENSITY.10.19
SCALE.1/24000
WINDOW.MINUTES.7.5.7.5
10.3.YES
TICK INTERVAL.MINUTES.7.5.7.5.MINUTES.2.5.2.5
ZONE
NEXT.ON.BATCH ... ON.ECHO
OFF.DEFAULT..PLTCLASS

**DAM PACKAGE APPENDIX J
DEFAULT COMMANDS**

**DEF-STATUS
001**

**ON.DEFAULT..STATUS
ON.PROMPT.CONFIRM
NEXT.ON.BATCH ... ON.ECHO
OFF.DEFAULT..STATUS**

PREFACE TO APPENDIX K

THE DAM PACKAGE SOFTWARE IS DESIGNED TO RUN ON UNIVAC 1100 COMPUTERS UNDER THE EXEC-8 OPERATING SYSTEM. THE PRIMARY IMPLEMENTATION LANGUAGE IS UNIVAC FORTRAN V, EXTENDED WITH STANDARD CONVENTIONS FOR SUBROUTINE INTERFACING AND EXTENSIVE LIBRARIES OF PSEUDO RECORD STRUCTURES AND UTILITY ROUTINES. THE SECONDARY IMPLEMENTATION LANGUAGES ARE UNIVAC 1100 ASSEMBLY LANGUAGE, EXEC-8 COMMAND LANGUAGE, AND EXEC-8 EXECUTIVE REQUESTS (ER'S). WHERE POSSIBLE, ALL MODULES ARE STRUCTURED AND DOCUMENTED IN SIMILAR FASHION, REGARDLESS OF THEIR IMPLEMENTATION LANGUAGE.

SUBROUTINE AND FUNCTION ARGUMENTS ARE ALWAYS DECLARED IN THE FOLLOWING STANDARD FORM:

1. EACH ARGUMENT (OR RELATED SET OF ARGUMENTS) IS DECLARED ON A SEPARATE LINE.
2. THE FORTRAN CONTINUATION CHARACTER (COLUMN 6) INDICATES WHETHER THE ARGUMENT IS OUTPUT 'O', INPUT 'I', UPDATE 'U', OR A POINTER '*' TO ANOTHER ARGUMENT.
3. EACH ARGUMENT IS EXPLAINED WITH AN INLINE COMMENT.
4. OUTPUT ARGUMENTS AND THEIR ASSOCIATED POINTERS ALWAYS PRECEDE INPUT ARGUMENTS AND THEIR ASSOCIATED POINTERS.
5. THE NAMING OF ARGUMENTS ALWAYS FOLLOWS THE FORTRAN NAME RULE (I THRU N ARE INTEGER).

LABELLED COMMON BLOCKS ARE ALWAYS MAINTAINED IN THE MACRO LIBRARY (APPENDIX Q) IN THE FOLLOWING STANDARD FORM:

1. EACH COMMON VARIABLE (OR RELATED SET COMMON VARIABLES) IS DECLARED ON A SEPARATE LINE.
2. EACH COMMON VARIABLE IS EXPLAINED WITH AN INLINE COMMENT.
3. THE NAMING OF COMMON VARIABLES ALWAYS FOLLOWS THE FORTRAN NAME RULE (I THRU N ARE INTEGER).

**DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION**

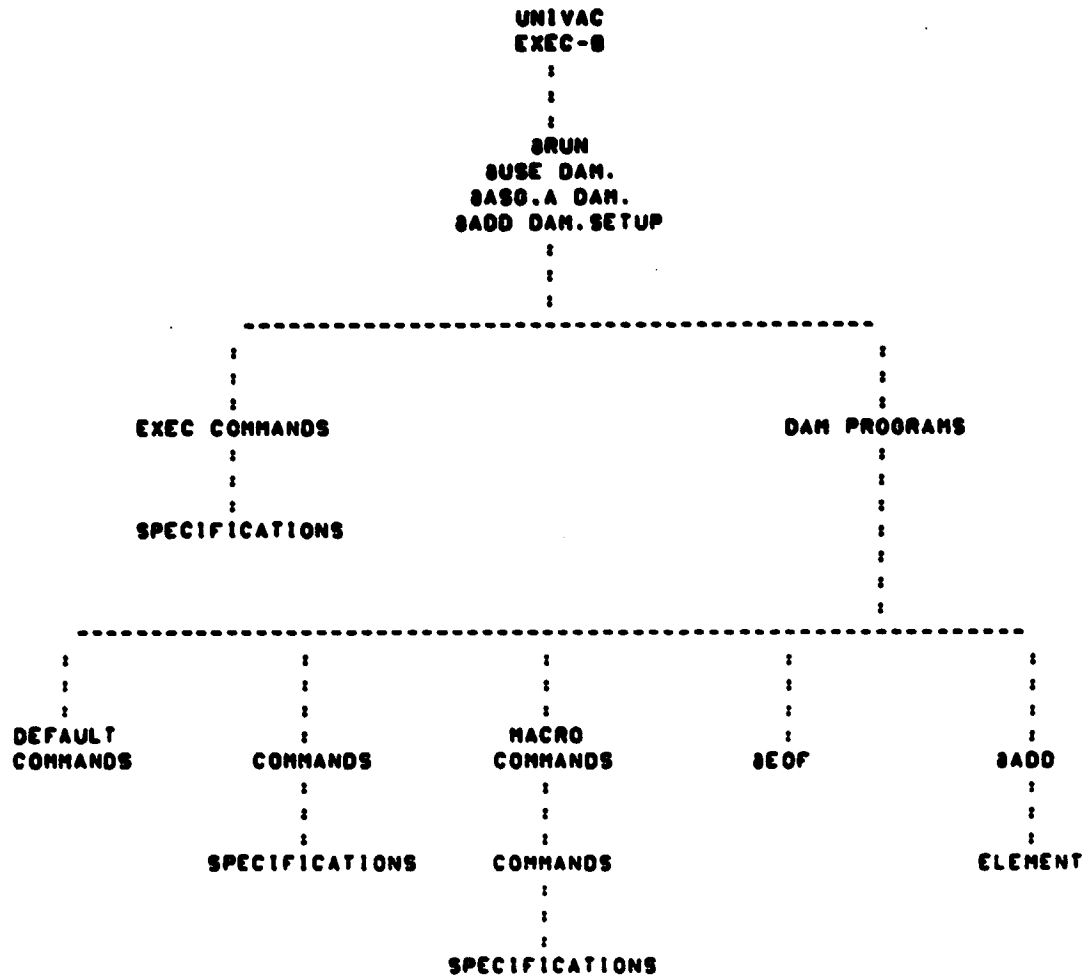
**APPENDIX-K
001**

**SPRT.SC DAM.PREFACE-K
SPRT.SC DAM.APPENDIX-K
SPRT.SC DAM.SYS-HIA
SMSG.N DAM.SYS-TITLE
SMSG.N DAM.SYS-LIST
SPRT.SC DAM.SYS-EXPLAIN
SPRT.SC DAM.SYS-COPYCOM
SPRT.SC DAM.SYS-FOROPT
SPRT.SC DAM.SYS-BLOCK
SPRT.SC DAM.SYS-MAPOPT
SPRT.SC DAM.SYS-COMPILE
SPRT.SC DAM.SYS-COLLECT
SPRT.SC DAM.SYS-DELETE
SPRT.SC DAM.SYS-GENCOM**

- . HIERARCHY**
- . TITLE PAGE**
- . PRINT ANNOTATED LIST OF SYMBOLIC ELEMENTS**
- . IMPLEMENTATION INSTRUCTIONS**
- . COPY COMPILATION/COLLECTION COMMAND STREAMS TO TFFS**
- . STANDARD FORTRAN COMPILER OPTIONS**
- . BLOCK DATA SUBROUTINE**
- . STANDARD MAP PROCESSOR OPTIONS**
- . COMPILATION COMMAND STREAM (MUST NOT BE ADDED FROM DAM)**
- . COLLECTION COMMAND STREAM (MUST NOT BE ADDED FROM DAM)**
- . DELETE SOURCE SYMBOLICS (MUST NOT BE ADDED FROM DAM)**
- . FOR JSC TO GENERATE DAM.SYS-COMPILE, DAM.SYS-DELETE**

**ORIGINAL PAGE IS
OF POOR QUALITY**

DAN PACKAGE HIERARCHY



SYSTEM IMPLEMENTATION (VERSION 0000)

(E H SCHLOSSER)

1.0 PERMANENT DISK FILES

THE FOLLOWING 3 PERMANENT CATALOGED PUBLIC DISK FILES ARE NORMALLY REQUIRED TO SUPPORT THE DAM PACKAGE:

PROGRAM FILE - READ ONLY, WRITE KEY, SECURE, 512 TRACKS MAXIMUM
USER FILE - SECURE (FOR USER-MAINTAINED CONTROL NETWORKS)
LOG FILE - SECURE (SEE PARAGRAPH 4.7)

2.0 COMPILATION AND COLLECTION

SYMBOLIC AND RELOCATABLE ELEMENTS FOR THE DAM PACKAGE ARE SUPPLIED ON TAPE IN UNIVAC EXEC 8 COPOUT FORMAT. THIS TAPE SHOULD BE LOADED ONTO A CATALOGED FILE ON DISK USING THE SCOPIN EXEC COMMAND.

2.1 TO GENERATE A REFERENCE LISTING OF ALL SYMBOLIC ELEMENTS IN THE PROGRAM FILE, INCLUDE THE FOLLOWING CARDS IN A BATCH OR DEMAND RUN:

```
...  
BUSE DAM.<QUALIFIER><FILE>  
(SPRT.T) (FIX FOR EXEC SYNC BUG)  
BASO.A DAM.  
BADD DAM.SYS-LIST  
...
```

2.2 THE DAM PACKAGE IS WRITTEN IN UNIVAC EXEC-8 CONTROL LANGUAGE, EXTENDED UNIVAC FORTRAN V (FIELDATA), AND UNIVAC 1100 ASSEMBLER. BECAUSE OF DIFFERENCES IN HARDWARE AND SYSTEMS SOFTWARE FROM ONE INSTALLATION TO ANOTHER, ALL SOURCE CODE SHOULD BE RE-COMPILED AND RE-MAPPED BEFORE ATTEMPTING TO EXECUTE ANY PROGRAMS. THE RUNSTREAM TO COMPILE AND COLLECT THE DAM PACKAGE IS AS FOLLOWS:

```
...  
BUSE DAM.<QUALIFIER><FILE>  
(SPRT.T) (FIX FOR EXEC SYNC BUG)  
BASO.A DAM.  
(LOCAL CHANGES TO DAM.SYS-COLLECT) (SEE 2.4)  
(LOCAL CHANGES TO MACROS) (SEE 5.2)  
(LOCAL CHANGES TO DAM.S'S-FOROPT) (SEE 4.1)  
BADD DAM.SYS-COPYCOM  
BADD COMPILE  
(BADD DELETE) (SEE 2.3)  
(LOCAL CHANGES TO DAM.SYS-BLOCK) (SEE 4.0)  
(LOCAL CHANGES TO DAM.SYS-MAPOPT) (SEE 4.1)  
BADD COLLECT  
(LOCAL CHANGES TO DAM.NEW-DAM) (SEE 4.8)  
(LOCAL CHANGES TO DEFAULT COMMANDS) (SEE 5.3)  
...
```

2.3 THE DAM.SYS-COPYCOM ELEMENT COPIES COMMAND STREAMS FROM THE DAM FILE TO

THE FOLLOWING 3 ELEMENTS IN TPF5 (THEY MUST NOT BE SADD-ED DIRECTLY FROM THE DAM FILE SINCE THEY SPACK IT):

- TPF5.COMPILE - CONTAINS SPOP, SDELETE.A, SFOR, SASH STATEMENTS, AS APPROPRIATE FOR ALL PROCS, PROGRAMS, SUBROUTINES.
- TPF5.DELETE - CONTAINS SDELETE.S STATEMENTS FOR ALL SOURCE PROGRAMS AND SUBROUTINES (OTHER SYMBOLIC ELEMENTS MUST NOT BE DELETED). ONCE ALL PROGRAMS ARE WORKING PROPERLY, THIS ELEMENT MAY BE SADD-ED TO REDUCE THE SIZE OF THE DAM PROGRAM FILE AND ENHANCE SECURITY.
- TPF5.COLLECT - COPIED FROM DAM.SYS-COLLECT (SEE BELOW)

2.4 THE DAM.SYS-COLLECT ELEMENT CONTAINS ONE SNAP STATEMENT FOR EVERY PROGRAM IN THE DAM PACKAGE. ANY SNAP STATEMENT SPECIFYING THE 'VIRTUAL' VERSION OF THE MAP SOURCE ELEMENT COLLECTS A VIRTUAL OR 'SKELETON' ABSOLUTE ELEMENT WHICH, WHEN SXQT-ED, TRIGGERS THE COLLECTION IN TPF5 AND EXECUTION OF THE REAL MAIN PROGRAM. THIS USER-TRANSPARENT TECHNIQUE IS USED FOR LESS FREQUENTLY EXECUTED PROGRAMS IN ORDER TO REDUCE THE SIZE OF THE DAM PROGRAM FILE. SNAP OPTION B MUST NOT BE USED SINCE SEVERAL PROGRAMS/SUBROUTINES ASSUME VARIABLES ARE INITIALLY ZERO.

3.0 EXECUTION

THE FIRST STATEMENTS OF EVERY DAM PACKAGE RUN SHOULD BE:

```
SRUN.<PRIORITY>/NR <RUNID, ETC.>  
SUSE DAM.<QUALIFIER>*<FILE>  
(SPRT.T) (FIX FOR EXEC SYNC BUG)  
SASO.A DAM.  
SADD DAM.SETUP
```

PROGRAMS SHOULD THEN BE EXECUTED FROM TPF5. THIS PROCEDURE WILL INSURE THAT THE 'VIRTUAL' VERSION OF A PROGRAM TRIGGERS THE COLLECTION OF THE REAL MAIN PROGRAM ONLY ONCE IN ANY RUN. NO MATTER HOW MANY TIMES THE PROGRAM IS SXQT-ED. THE RUN CARD FOR EACH SEPARATE RUN MUST HAVE A UNIQUE SIX-CHARACTER RUNID, AND A PROJECT-ID SHORTER THAN 12 CHARACTERS (UNLESS UNIQUE). THIS WILL HELP USERS KEEP TRACK OF OUTPUT AND ENSURE UNIQUE QUALIFIERS FOR WORK FILES, AS EXPLAINED IN PARAGRAPH 6. IF PARITY ERRORS OCCUR IN READING ORIGINAL ERTS TAPES, COPIES SHOULD BE MADE WITH DAM.ERTS-DUP, AND THE COPIES USED. PROGRAMS IN A REMOTE BATCH RUN MAY USE THE 'PRINTER' COMMAND TO ROUTE ALTERNATE PRINT FILES BACK TO THE REMOTE SITE AND DEFINE THE CHARACTERISTICS OF THE REMOTE PRINTER, IF DESIRED.

4.0 REQUIRED LOCAL MODIFICATIONS

MODIFICATIONS REQUIRED AT DIFFERENT INSTALLATIONS:

4.1 THE COMPILE & COLLECT OPTIONS SPECIFIED IN DAM.SYS-FOROPT & DAM.SYS-MAPOPT MUST BE COMPATIBLE WITH LOCAL HARDWARE AND SOFTWARE. IF THE UNIVAC FORTRAN V (FIELDATA) LIBRARY ROUTINES ARE NOT IN THE SYSTEM RELOCATABLE LIBRARY, AND/OR THE UNIVAC MATHPACK ROUTINES ARE NOT IN THE SYSTEM RELOCATABLE LIBRARY, THEN CODE SIMILAR TO THE FOLLOWING MUST BE ADDED TO DAM.SYS-MAPOPT:

```
LIB DAM.  
LIB <NAME OF FILE WITH FORTRAN V (FIELDATA) LIBRARY RELOCATABLES>  
LIB <NAME OF FILE WITH MATHPACK RELOCATABLES>  
LIB DAM.
```

LIB <NAME OF FILE WITH FORTRAN V (FIELDATA) LIBRARY RELOCATABLES>
LIB <NAME OF FILE WITH MATHPACK RELOCATABLES>

- 4.2 THE DAM PACKAGE REQUIRES AT LEAST ONE ALTERNATE PRINT FILE. IF MORE ARE AVAILABLE (UP TO A MAXIMUM OF 9) IT WILL USE THEM TO MINIMIZE THE DISK ACCESSES REQUIRED FOR MAPS MORE THAN ONE LINE-PRINTER PAGE IN WIDTH. THE VALUE ASSIGNED TO MSAITH IN DAM.SYS-BLOCK MUST ALWAYS EQUAL THE INSTALLATION EXEC-8 SYSTEM GENERATION PARAMETER SMAITH (MAXIMUM ALLOWABLE NUMBER OF ACTIVE ALTERNATE PRINT FILES)!!! THE VALUE ASSIGNED TO MALTH IN DAM.SYS-BLOCK REPRESENTS THE DEFAULT NUMBER OF ALTERNATE PRINT FILES TO BE USED BY PROGRAMS IN THE DAM PACKAGE. MALTH SHOULD GENERALLY CORRESPOND TO THE WIDTH IN LINE-PRINTER PAGES OF THE WIDEST MAP THAT NORMALLY WILL BE GENERATED (4 IS TYPICAL). HOWEVER, MALTH MUST NEVER EXCEED 9 AND IT MUST NEVER EXCEED MSAITH!! (THE 'PRINTER' COMMAND CAN BE USED WITHIN A PROGRAM TO TEMPORARILY CHANGE MALTH BUT NOT MSAITH.)
- 4.3 THE DEVICE-TYPE-MNEMONIC USED TO DESIGNATE ONSITE LINE PRINTER(S) SHOULD BE ASSIGNED TO MNEHON IN DAM.SYS-BLOCK. (THE PRINTER COMMAND CAN BE USED WITHIN A PROGRAM TO DYNAMICALLY RE-ROUTE ALTERNATE PRINT FILES TO A REMOTE PRINTER AND TO DEFINE ITS CHARACTERISTICS.)
- 4.4 LSINCH (SYSTEM PRINT LINES PER INCH) AND LSPACE (SYSTEM PRINT LINES PER PAGE) IN DAM.SYS-BLOCK MUST EXACTLY MATCH THE INSTALLATION STANDARDS!!
- 4.5 IT IS HIGHLY DESIREABLE (BUT NOT NECESSARY) THAT DAM PACKAGE ALTERNATE PRINT FILES BE PRODUCED AT 8 LINES PER INCH TO PROVIDE MAXIMUM RESOLUTION FOR THE COMPUTER-GENERATED MAPS. LINC (DAM PRINT LINES PER INCH) AND LPAGE (DAM PRINT LINES PER PAGE) AS SPECIFIED IN DAM.SYS-BLOCK NEED NOT NECESSARILY MATCH THE INSTALLATION STANDARDS. IF THEY DO MATCH THE DEFAULT CHARACTERISTICS OF THE PRINTER(S) SPECIFIED BY MNEHON (SEE 4.3) THEN THE VALUE ASSIGNED TO KONPRT IN DAM.SYS-BLOCK MUST BE 'NON'. OTHERWISE THE VALUE MUST BE 'MAN' FOR PRINTERS WHERE THE LINES PER INCH ARE MANUALLY CONTROLLED BY AN OPERATOR AND 'AUT' FOR PRINTERS WHERE THE LINES PER INCH ARE AUTOMATICALLY CONTROLLED BY THE SOFTWARE.
- 4.6 KINC (DAM PRINT COLUMNS PER INCH) AND KPAGE (DAM PRINT COLUMNS PER PAGE) IN DAM.SYS-BLOCK SHOULD MATCH THE STANDARDS OF THE PRINTER(S) SPECIFIED BY MNEHON (SEE 4.3).
- 4.7 ALL PROGRAMS AUTOMATICALLY MAKE ENTRIES IN A LOG FILE WHICH MAY BE SELECTIVELY QUERIED BY THE STATUS PROGRAM. THE <QUALIFIER><FILE> AND FILE SIZE IN SECTORS OF THE LOG FILE, SPECIFIED BY LOOFIL AND LONSEC, MUST BE ASSIGNED IN DAM.SYS-BLOCK. IF LONSEC IS LESS THAN 100 NO LOG IS KEPT. OTHERWISE THE LOG FILE IS AUTOMATICALLY CATALOGED, INITIALIZED, AND MAINTAINED BY THE DAM PACKAGE. (THE SYSTEMS ANALYST MUST NOT CATALOG IT.)
- 4.8 DAM.NEW-DAM SHOULD BE REVISED AS APPROPRIATE, SUBJECT TO THE FOLLOWING: THIS NEWS ELEMENT MUST BE PRESENT, EVEN IF EMPTY. THE FIRST CHARACTER OF EACH LINE IS A FORTRAN PRINT CONTROL CHARACTER. LINES SHOULD NOT BE LONGER THAN 80 CHARACTERS. THIS ELEMENT MAY BE UPDATED AT ANY TIME WITHOUT RECOMPILING.

5.0 OPTIONAL LOCAL MODIFICATIONS

MODIFICATIONS WHICH MAY BE DESIRABLE FOR DIFFERENT INSTALLATIONS/APPLICATIONS:

5.1 THE DAN PACKAGE GENERATES BOX PAGES CONTAINING RUNID, PROGRAM, DATE, TIME, ETC. ON THE FRONT OF EACH ALTERNATE PRINT FILE, BUT NOT ON THE STANDARD PRINTS FILE. INSTALLATIONS WHICH DO NOT AUTOMATICALLY GENERATE BOX PAGES FOR THE PRINTS FILE SHOULD HAVE USERS SKGT A LOCAL BOX PAGE UTILITY PROGRAM AT THE BEGINNING OF EACH BATCH RUN.

5.2 TWO ARRAYS MAY NOT BE THE OPTIMUM SIZES FOR ALL INSTALLATIONS AND APPLICATIONS:

KTABLE - THIS ARRAY STORES TICK DATA FOR EACH MAP, PACKED ONE TICK PER WORD. IF MAPS ARE VERY LARGE OR CONTAIN CLOSELY SPACED TICKS THIS ARRAY MAY HAVE TO BE INCREASED IN SIZE. TO DO THIS CHANGE THE KTBLSZ PARAMETER STATEMENT IN THE KOMTBL PROC (ELEMENT FORPROC).

KSYN - THIS ARRAY STORES CHARACTER SYMBOLS FOR EACH MAP, ONE SYMBOL GROUP PER WORD. IF MAP SCALES MUCH SMALLER THAN 1/250,000 ARE TO BE USED THIS ARRAY MUST BE ENLARGED. TO DO THIS CHANGE THE KSYMSZ PARAMETER STATEMENT IN KOSYN-PROC (APPENDIX-Q).

5.3 MOST PROGRAMS IN THE DAN PACKAGE USE DEFAULT COMMANDS CONTAINED IN SYMBOLIC ELEMENTS NAMED AS FOLLOWS:

DAM.DEF-(PROGRAM NAME)

DEFAULT COMMANDS MAY BE CHANGED WITHOUT RECOMPILING.

6.0 CONFLICTS BETWEEN RUNS

SEVERAL PROGRAMS IN THE DAN PACKAGE DYNAMICALLY ASSIGN CATALOGED P4STRAND FILES. THESE FILES ARE USED TO ALLOW FOR DATA TRANSFER BETWEEN PROGRAMS IN DIFFERENT RUNS, TO MAXIMIZE RERUN EFFICIENCY AFTER IRRECOVERABLE TAPE PARITY ERROR OR SYSTEM CRASH, AND FOR ALTERNATE PRINT FILES. THE FOLLOWING EXTERNAL FILE NAMES ARE USED FOR THESE FILES:

***DAMPRT-0, THROUGH *DAMPRT-9.**

***DANDET-1, THROUGH *DANDET-4.**

TO PREVENT CONFLICTS BETWEEN CONCURRENT RUNS THE QUALIFIER USED FOR THESE FILES MUST BE UNIQUE FOR EACH RUN. TO INSURE THIS UNIQUENESS, THE DAN PACKAGE AUTOMATICALLY EXTENDS INITIAL USER-SPECIFIED QUALIFIERS SHORTER THAN 12 CHARACTERS WITH NON-BLANK CHARACTERS FROM THE RIGHT OF THE USER-SPECIFIED RUNID. IF THE EXTENDED QUALIFIER IS NOT UNIQUE, ERROR TERMINATION MAY RESULT.

7.0 DEBUGGING

7.1 THE NUMBER OF COPIES SPECIFIED WITH THE 'COPIES' COMMAND SHOULD NORMALLY BE BETWEEN 1 AND 9. IF 0 IS SPECIFIED, ALTERNATE PRINT FILES ARE NOT BSYM-ED INTERNALLY. THE SYSTEMS ANALYST MAY THEN EXAMINE THESE FILES FROM A REMOTE TERMINAL WITH THE SED PROCESSOR AND MUST MANUALLY BSYM OR BDELETE THEM.

7.2 THE 'ON' AND 'OFF' COMMANDS MAY BE USED TO CONTROL TRACING & JUMPING.

7.3 THE 'PEEK' AND 'POKE' COMMANDS MAY BE USED TO INSPECT AND TO MODIFY

(AT THE ANALYST'S PERIL) VARIABLES IN LABELLED COMMON BLOCKS.

7.4 PICTAB AND PRTCLASS ASSIGN SEVERAL ALTERNATE PRINT FILES ON DISK. IF A SINGLE ALTERNATE PRINT FILE ON TAPE IS DESIRED INSTEAD, THEN THE FOLLOWING CARD MUST APPEAR IN THE RUN BEFORE THESE PROGRAMS ARE 3XQT-ED:
3ASO.<OPTIONS> 10..09.<REEL NUMBER>

7.5 THE DAM SOFTWARE INCLUDES NUMEROUS FIXES FOR BUGS IN RECENT RELEASES OF UNIVAC SYSTEMS SOFTWARE. ALL PROGRAMS AND SUBROUTINES IN THE DAM PACKAGE HAVE BEEN SUCCESSFULLY COMPILED, COLLECTED, AND EXECUTED ON THE UNIVAC 1110 AT NASA JOHNSON SPACE CENTER UNDER THE FOLLOWING SYSTEMS SOFTWARE:

EXEC 31.244.211B
FURPUR 0026
POPT10 RL70-6
FOR SOE3
ASH13B RL1869
MAP27.1 RL71-3
SYSS*RL18S. LEVEL 71-3

IT IS NOT POSSIBLE TO INCLUDE FIXES FOR BUGS IN ALL PAST AND FUTURE UNIVAC SYSTEMS RELEASES. IF BUGS IN THE SYSTEM PROCESSORS AND/OR SYSTEM RELOCATABLE LIBRARY AFFECT THE DAM PACKAGE, THEY CAN ORDINARILY BE CIRCUMVENTED BY GOING TO AN EARLIER OR LATER SYSTEMS RELEASE.

8.0 MONITORING USAGE

THE STATUS PROGRAM MAY BE USED (EITHER IN BATCH OR DEMAND) TO LIST ALL RUNS USING THE DAM PACKAGE. ORDINARILY, STATUS WILL NOT PRINT USER ACCOUNT NUMBERS, AND WILL NOT LIST MORE THAN 20 RUNS. HOWEVER, THE FOLLOWING RUNSTREAM AVOIDS BOTH THESE LIMITATIONS:

3XQT STATUS
KEY.<WRITE KEY FOR THE DAM PROGRAM FILE>
///// .9999
EXIT

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COPYCON
001

SCOPY.S DAM.SYS-COMPILE.COMPILE
SCOPY.S DAM.SYS-DELETE.DELETE
SCOPY.S DAM.SYS-COLLECT.COLLECT

. ADD ELEMENT MUST NOT PACK ITS OWN FILE!!!
. ADD ELEMENT MUST NOT PACK ITS OWN FILE!!!
. ADD ELEMENT MUST NEVER PACK ITS OWN FILE!!

DAH PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-FOROPT
001

COMPILER (DATA-SHORT).(RETN-NOP).(1110-OPT)
SEOF

DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-NAPOPT
001

TYPE CLRAFCH
CLASS 1110
CLASS ASH
SEOF

3H00 DAM FORTRAN PROCEDURE FORPROCS/
3PDP.LF DAM.FORPROCS/..FORPROCS/
3H00 DAM FORTRAN PROCEDURE TRFORM-PROCS/
3PDP.LF DAM.TRFORM-PROCS/..TRFORM-PROCS/
3H00 DAM FORTRAN PROCEDURE PICDEF-PROC/
3PDP.LF DAM.PICDEF-PROC/..PICDEF-PROC/
3H00 DAM FORTRAN PROCEDURE MAXBYT-PROC/
3PDP.LF DAM.MAXBYT-PROC/..MAXBYT-PROC/
3H00 DAM FORTRAN PROCEDURE MAXICE-PROC/
3PDP.LF DAM.MAXICE-PROC/..MAXICE-PROC/
3H00 DAM FORTRAN PROCEDURE MAXINT-PROC/
3PDP.LF DAM.MAXINT-PROC/..MAXINT-PROC/
3H00 DAM FORTRAN PROCEDURE NULCHR-PROC/
3PDP.LF DAM.NULCHR-PROC/..NULCHR-PROC/
3H00 DAM FORTRAN PROCEDURE NULCST-PROC/
3PDP.LF DAM.NULCST-PROC/..NULCST-PROC/
3H00 DAM FORTRAN PROCEDURE ASHDEF-PROC/
3PDP.LF DAM.ASHDEF-PROC/..ASHDEF-PROC/
3H00 DAM FORTRAN PROCEDURE FIDEF-PROC/
3PDP.LF DAM.FIDEF-PROC/..FIDEF-PROC/
3H00 DAM FORTRAN PROCEDURE FACBIT-PROC/
3PDP.LF DAM.FACBIT-PROC/..FACBIT-PROC/
3H00 DAM FORTRAN PROCEDURE KOMLUS-PROC/
3PDP.LF DAM.KOMLUS-PROC/..KOMLUS-PROC/
3H00 DAM FORTRAN PROCEDURE KOMIO-PROC/
3PDP.LF DAM.KOMIO-PROC/..KOMIO-PROC/
3H00 DAM FORTRAN PROCEDURE PXBDEF-PROC/
3PDP.LF DAM.PXBDEF-PROC/..PXBDEF-PROC/
3H00 DAM FORTRAN PROCEDURE CBDEF-PROC/
3PDP.LF DAM.CBDEF-PROC/..CBDEF-PROC/
3H00 DAM FORTRAN PROCEDURE KOMSLM-PROC/
3PDP.LF DAM.KOMSLM-PROC/..KOMSLM-PROC/
3H00 DAM FORTRAN PROCEDURE PRCDEF-PROC/
3PDP.LF DAM.PRCDEF-PROC/..PRCDEF-PROC/
3H00 DAM FORTRAN PROCEDURE ALTPRT-PROCS/
3PDP.LF DAM.ALTPRT-PROCS/..ALTPRT-PROCS/
3H00 DAM FORTRAN PROCEDURE KOMIRT-PROC/
3PDP.LF DAM.KOMIRT-PROC/..KOMIRT-PROC/
3H00 DAM FORTRAN PROCEDURE KOMNET-PROC/
3PDP.LF DAM.KOMNET-PROC/..KOMNET-PROC/
3H00 DAM FORTRAN PROCEDURE LSTLUB-PROC/
3PDP.LF DAM.LSTLUB-PROC/..LSTLUB-PROC/
3H00 DAM FORTRAN PROCEDURE WINDOW-PROCS/
3PDP.LF DAM.WINDOW-PROCS/..WINDOW-PROCS/
3H00 DAM ASSEMBLER PROCEDURE GETOPT-APROC/
3PDP.L DAM.GETOPT-APROC/..GETOPT-APROC/
3H00 DAM ASSEMBLER PROCEDURE KOMXQT-APROC/
3PDP.L DAM.KOMXQT-APROC/..KOMXQT-APROC/
3H00 DAM FORTRAN PROCEDURE NERDET-PROCS/
3PDP.LF DAM.NERDET-PROCS/..NERDET-PROCS/
3H00 DAM FORTRAN PROCEDURE PRODEF-PROC/
3PDP.LF DAM.PRODEF-PROC/..PRODEF-PROC/
3H00 DAM FORTRAN PROCEDURE KOML2N-PROC/
3PDP.LF DAM.KOML2N-PROC/..KOML2N-PROC/
3H00 DAM FORTRAN PROCEDURE KOMKS-PROC/
3PDP.LF DAM.KOMKS-PROC/..KOMKS-PROC/

ORIGINAL PAGE IS
OF POOR QUALITY

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
002

```

#HDD DAM FORTRAN PROCEDURE KONSYN-PROC/
#POD.LF DAM.KONSYN-PROC/..KONSYN-PROC/
#HDD DAM FORTRAN PROCEDURE KOHLU3-PROC/
#POD.LF DAM.KOHLU3-PROC/..KOHLU3-PROC/
#HDD DAM FORTRAN PROCEDURE XQTLOG-PROCS/
#POD.LF DAM.XQTLOG-PROCS/..XQTLOG-PROCS/
#HDD DAM FORTRAN PROCEDURE KOMTBL-PROC/
#POD.LF DAM.KOMTBL-PROC/..KOMTBL-PROC/
#PACK.SR DAM.
#HDD DAM PICTAB-MAP/VIRTUAL
#HDD DAM STATUS-MAP/
#HDD DAM ERTSIDC-MAP/
#HDD DAM ERTSIDC-MAP/VIRTUAL
#HDD DAM CLASSIFY-MAP/VIRTUAL
#HDD DAM PRCLCLASS-MAP/VIRTUAL
#HDD DAM PLTCLASS-MAP/VIRTUAL
#HDD DAM STATUS-MAP/VIRTUAL
#HDD DAM ERPRCN/DAM
#ASH.FS DAM.ERPRCN/DAM
#HDD DAM CONTROL-MAP/VIRTUAL
#HDD DAM LOCATE-MAP/
#HDD DAM LOCATE-MAP/VIRTUAL
#HDD DAM DLSTSQ/
#FOR.S DAM.DLSTSQ/
#ADD DAM.SYS-FOROPT
#HDD DAM ERSPRTCN-MAP/
#HDD DAM ERSPRTCN-MAP/VIRTUAL
#HDD DAM DIAERR/
#FOR.S DAM.DIAERR/
#ADD DAM.SYS-FOROPT
#HDD DAM GAPFOC/
#FOR.S DAM.GAPFOC/
#ADD DAM.SYS-FOROPT
#HDD DAM GAPLUF/
#FOR.S DAM.GAPLUF/
#ADD DAM.SYS-FOROPT
#HDD DAM GAPLUR/
#FOR.S DAM.GAPLUR/
#ADD DAM.SYS-FOROPT
#HDD DAM FOCUS/
#FOR.S DAM.FOCUS/
#ADD DAM.SYS-FOROPT
#HDD DAM NULSUB/
#ASH.FS DAM.NULSUB/
#HDD DAM DMPTIC/
#FOR.S DAM.DMPTIC/
#ADD DAM.SYS-FOROPT
#HDD DAM PICPR3/
#FOR.S DAM.PICPR3/
#ADD DAM.SYS-FOROPT
#HDD DAM ERERR/DAM
#ASH.FS DAM.ERERR/DAM
#HDD DAM EREXIT/DAM
#ASH.FS DAM.EREXIT/DAM
#HDD DAM PIC015/
#FOR.S DAM.PIC015/

```

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
003

3ADD DAM.SYS-FOROPT
3HDD DAM PICP46/
3FOR.S DAM.PICP46/
3ADD DAM.SYS-FOROPT
3HDD DAM 184350/
3ASH.FS DAM.184350/
3HDD DAM PICL15/
3FOR.S DAM.PICL15/
3ADD DAM.SYS-FOROPT
3HDD DAM USWAP-MAP/
3HDD DAM USWAP-MAP/VIRTUAL
3HDD DAM DATA-MAP/
3HDD DAM DATA-MAP/VIRTUAL
3HDD DAM PICP49/
3FOR.S DAM.PICP49/
3ADD DAM.SYS-FOROPT
3HDD DAM PICPR9/
3FOR.S DAM.PICPR9/
3ADD DAM.SYS-FOROPT
3HDD DAM PICT03/
3FOR.S DAM.PICT03/
3ADD DAM.SYS-FOROPT
3HDD DAM PICT09/
3FOR.S DAM.PICT09/
3ADD DAM.SYS-FOROPT
3HDD DAM CLADE5/
3FOR.S DAM.CLADE5/
3ADD DAM.SYS-FOROPT
3PACK DAM.
3HDD DAM CLADE9/
3FOR.S DAM.CLADE9/
3ADD DAM.SYS-FOROPT
3HDD DAM OPRCLA/
3FOR.S DAM.OPRCLA/
3ADD DAM.SYS-FOROPT
3HDD DAM SLMCLA/
3FOR.S DAM.SLMCLA/
3ADD DAM.SYS-FOROPT
3HDD DAM COPYPX/
3FOR.S DAM.COPYPX/
3ADD DAM.SYS-FOROPT
3HDD DAM PIC129/
3FOR.S DAM.PIC129/
3ADD DAM.SYS-FOROPT
3HDD DAM PIC345/
3FOR.S DAM.PIC345/
3ADD DAM.SYS-FOROPT
3HDD DAM PIC678/
3FOR.S DAM.PIC678/
3ADD DAM.SYS-FOROPT
3HDD DAM CLA129/
3FOR.S DAM.CLA129/
3ADD DAM.SYS-FOROPT
3HDD DAM CLA345/
3FOR.S DAM.CLA345/
3ADD DAM.SYS-FOROPT

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
004

SHDD DAM H2TOMI/
SFOR.S DAM.H2TOMI/
SADD DAM.SYS-FOROPT
SHDD DAM HILLBL/
SFOR.S DAM.HILLBL/
SADD DAM.SYS-FOROPT
SHDD DAM IDERT/
SFOR.S DAM.IDERT/
SADD DAM.SYS-FOROPT
SHDD DAM INVORI/
SFOR.S DAM.INVORI/
SADD DAM.SYS-FOROPT
SHDD DAM INVWIN/
SFOR.S DAM.INVWIN/
SADD DAM.SYS-FOROPT
SHDD DAM ISRTBA/
SFOR.S DAM.ISRTBA/
SADD DAM.SYS-FOROPT
SHDD DAM ISRTBD/
SFOR.S DAM.ISRTBD/
SADD DAM.SYS-FOROPT
SHDD DAM KSPRED/
SFOR.S DAM.KSPRED/
SADD DAM.SYS-FOROPT
SHDD DAM LSPRED/
SFOR.S DAM.LSPRED/
SADD DAM.SYS-FOROPT
SHDD DAM MSPRED/
SFOR.S DAM.MSPRED/
SADD DAM.SYS-FOROPT
SHDD DAM NEOPIC/
SFOR.S DAM.NEOPIC/
SADD DAM.SYS-FOROPT
SHDD DAM PRHIPX/
SFOR.S DAM.PRHIPX/
SADD DAM.SYS-FOROPT
SPACK DAM.
SHDD DAM PRH2PX/
SFOR.S DAM.PRH2PX/
SADD DAM.SYS-FOROPT
SHDD DAM QUAD/
SFOR.S DAM.QUAD/
SADD DAM.SYS-FOROPT
SHDD DAM RITADD/
SFOR.S DAM.RITADD/
SADD DAM.SYS-FOROPT
SHDD DAM RL2ISX/
SFOR.S DAM.RL2ISX/
SADD DAM.SYS-FOROPT
SHDD DAM RL2SX/
SFOR.S DAM.RL2SX/
SADD DAM.SYS-FOROPT
SHDD DAM RL4SX/
SFOR.S DAM.RL4SX/
SADD DAM.SYS-FOROPT
SHDD DAM ROTCHX/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
005

3FOR.S DAM.ROTCNX/
3ADD DAM.SYS-FOROPT
3HDD DAM ROTCOL/
3FOR.S DAM.ROTCOL/
3ADD DAM.SYS-FOROPT
3HDD DAM ROTROW/
3FOR.S DAM.ROTROW/
3ADD DAM.SYS-FOROPT
3HDD DAM SHFTBC/
3ASH.FS DAM.SHFTBC/
3HDD DAM SSPR/
3FOR.S DAM.SSPR/
3ADD DAM.SYS-FOROPT
3HDD DAM SUBH1/
3FOR.S DAM.SUBH1/
3ADD DAM.SYS-FOROPT
3HDD DAM VALKEY/
3FOR.S DAM.VALKEY/
3ADD DAM.SYS-FOROPT
3HDD DAM VALLBL/
3FOR.S DAM.VALLBL/
3ADD DAM.SYS-FOROPT
3HDD DAM HRVERT/
3FOR.S DAM.HRVERT/
3ADD DAM.SYS-FOROPT
3HDD DAM EAPRNT/DAM
3ASH.FS DAM.EAPRNT/DAM
3HDD DAM EAREAD/DAM
3ASH.FS DAM.EAREAD/DAM
3HDD DAM ERDATE/DAM
3ASH.FS DAM.ERDATE/DAM
3HDD DAM ERFACL/DAM
3ASH.FS DAM.ERFACL/DAM
3HDD DAM ERFITH/DAM
3ASH.FS DAM.ERFITH/DAM
3HDD DAM ERPCHA/DAM
3ASH.FS DAM.ERPCHA/DAM
3PACK DAM.
3HDD DAM ERPCT/DAM
3ASH.FS DAM.ERPCT/DAM
3HDD DAM ERPRCA/DAM
3ASH.FS DAM.ERPRCA/DAM
3HDD DAM ERPRNT/DAM
3ASH.FS DAM.ERPRNT/DAM
3HDD DAM ERPRTA/DAM
3ASH.FS DAM.ERPRTA/DAM
3HDD DAM ERREAD/DAM
3ASH.FS DAM.ERREAD/DAM
3HDD DAM ERREDA/DAM
3ASH.FS DAM.ERREDA/DAM
3HDD DAM ERSUPS/DAM
3ASH.FS DAM.ERSUPS/DAM
3HDD DAM ERTWAT/DAM
3ASH.FS DAM.ERTWAT/DAM
3HDD DAM PICPRO/
3FOR.S DAM.PICPRO/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILC
008

3ADD DAM.SYS-FOROPT
3HDD DAM PICTOT/
3FOR.S DAM.PICTOT/
3ADD DAM.SYS-FOROPT
3HDD DAM KMOGAS/
3FOR.S DAM.KMOGAS/
3ADD DAM.SYS-FOROPT
3HDD DAM KMOGFR/
3FOR.S DAM.KMOGFR/
3ADD DAM.SYS-FOROPT
3HDD DAM KHRVIN/
3FOR.S DAM.KHRVIN/
3ADD DAM.SYS-FOROPT
3HDD DAM ERPFS/DAM
3ASH.FS DAM.ERPFS/DAM
3HDD DAM DITCOP-MAP/
3HDD DAM DITVER/
3FOR.S DAM.DITVER/
3ADD DAM.SYS-FOROPT
3HDD DAM REVERT/
3FOR.S DAM.REVERT/
3ADD DAM.SYS-FOROPT
3HDD DAM DITCOP-MAP/VIRTUAL
3HDD DAM LOCDSF/
3FOR.S DAM.LOCDSF/
3ADD DAM.SYS-FOROPT
3HDD DAM KMOGAD/
3FOR.S DAM.KMOGAD/
3ADD DAM.SYS-FOROPT
3HDD DAM KMOGLO/
3FOR.S DAM.KMOGLO/
3ADD DAM.SYS-FOROPT
3HDD DAM LENPAD/
3FOR.S DAM.LENPAD/
3ADD DAM.SYS-FOROPT
3HDD DAM LOWCST/
3FOR.S DAM.LOWCST/
3ADD DAM.SYS-FOROPT
3PACK DAM.
3HDD DAM SREADS/
3FOR.S DAM.SREADS/
3ADD DAM.SYS-FOROPT
3HDD DAM SPANS/
3FOR.S DAM.SPANS/
3ADD DAM.SYS-FOROPT
3HDD DAM GETOKM/
3FOR.S DAM.GETOKM/
3ADD DAM.SYS-FOROPT
3HDD DAM NEXTOK/
3FOR.S DAM.NEXTOK/
3ADD DAM.SYS-FOROPT
3HDD DAM GETS/
3FOR.S DAM.GETS/
3ADD DAM.SYS-FOROPT
3HDD DAM ERIO/DAM
3ASH.FS DAM.ERIO/DAM

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
087

SHDD DAM ERION/DAM
SASH.FS DAM.ERION/DAM
SHDD DAM ERWAIT/DAM
SASH.FS DAM.ERWAIT/DAM
SHDD DAM TRUAL/
SFOR.S DAM.TRUAL/
SADD DAM.SYS-FOROPT
SHDD DAM TRUCST/
SFOR.S DAM.TRUCST/
SADD DAM.SYS-FOROPT
SHDD DAM KMOOBR/
SFOR.S DAM.KMOOBR/
SADD DAM.SYS-FOROPT
SHDD DAM QUARTN/
SFOR.S DAM.QUARTN/
SADD DAM.SYS-FOROPT
SHDD DAM QUARTU/
SFOR.S DAM.QUARTU/
SADD DAM.SYS-FOROPT
SHDD DAM VARSQU/
SFOR.S DAM.VARSQU/
SADD DAM.SYS-FOROPT
SHDD DAM VARSQN/
SFOR.S DAM.VARSQN/
SADD DAM.SYS-FOROPT
SHDD DAM SYSADD/DAM
SFOR.S DAM.SYSADD/DAM
SADD DAM.SYS-FOROPT
SHDD DAM KHXOS/
SFOR.S DAM.KHXOS/
SADD DAM.SYS-FOROPT
SHDD DAM CALSCA/
SFOR.S DAM.CALSCA/
SADD DAM.SYS-FOROPT
SHDD DAM KMDCOU/
SFOR.S DAM.KMDCOU/
SADD DAM.SYS-FOROPT
SHDD DAM DITXQT/
SFOR.S DAM.DITXQT/
SADD DAM.SYS-FOROPT
SHDD DAM SUBIN/
SFOR.S DAM.SUBIN/
SADD DAM.SYS-FOROPT
SPACK DAM.
SHDD DAM PRCEXI/
SFOR.S DAM.PRCEXI/
SADD DAM.SYS-FOROPT
SHDD DAM PLTCLASS/
SFOR.S DAM.PLTCLASS/
SADD DAM.SYS-FOROPT
SHDD DAM PLTCLASS-HAP/
SHDD DAM DITCOP/
SFOR.S DAM.DITCOP/
SADD DAM.SYS-FOROPT
SHDD DAM DITEXI/
SFOR.S DAM.DITEXI/

DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
000

8ADD DAN.SYS-FOROPT
8MDD DAN CONEXI/
8FOR.S DAN.CONEXI/
8ADD DAN.SYS-FOROPT
8MDD DAN PLCXQT/
8FOR.S DAN.PLCXQT/
8ADD DAN.SYS-FOROPT
8MDD DAN KNDPLO/
8FOR.S DAN.KNDPLO/
8ADD DAN.SYS-FOROPT
8MDD DAN LCSTEQ/
8FOR.S DAN.LCSTEQ/
8ADD DAN.SYS-FOROPT
8MDD DAN PLCHAP/
8FOR.S DAN.PLCHAP/
8ADD DAN.SYS-FOROPT
8MDD DAN PSTOP/
8FOR.S DAN.PSTOP/
8ADD DAN.SYS-FOROPT
8MDD DAN CLARAD/
8FOR.S DAN.CLARAD/
8ADD DAN.SYS-FOROPT
8MDD DAN PICPA4/
8FOR.S DAN.PICPA4/
8ADD DAN.SYS-FOROPT
8MDD DAN PICPA3/
8FOR.S DAN.PICPA3/
8ADD DAN.SYS-FOROPT
8MDD DAN PICPA2/
8FOR.S DAN.PICPA2/
8ADD DAN.SYS-FOROPT
8MDD DAN PICPA1/
8FOR.S DAN.PICPA1/
8ADD DAN.SYS-FOROPT
8MDD DAN PICLI4/
8FOR.S DAN.PICLI4/
8ADD DAN.SYS-FOROPT
8MDD DAN PICDI9/
8FOR.S DAN.PICDI9/
8ADD DAN.SYS-FOROPT
8MDD DAN PICLI9/
8FOR.S DAN.PICLI9/
8ADD DAN.SYS-FOROPT
8MDD DAN LOOZ/
8ASH.FS DAN.LOOZ/
8MDD DAN NTABS/DAN
8ASH.FS DAN.NTABS/DAN
8MDD DAN CBINIT/
8FOR.S DAN.CBINIT/
8ADD DAN.SYS-FOROPT
8MDD DAN WARNS/
8FOR.S DAN.WARNS/
8ADD DAN.SYS-FOROPT
8MDD DAN.
8MDD DAN IDUP/
8ASH.FS DAN.IDUP/
8MDD DAN LENCST/
8FOR.S DAN.LENCST/
8ADD DAN.SYS-FOROPT

DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
000

SHDD DAN PSTART/
SFOR.S DAN.PSTART/
SADD DAN.SYS-FOROPT
SHDD DAN READS/
SFOR.S DAN.READS/
SADD DAN.SYS-FOROPT
SHDD DAN PICD14/
SFOR.S DAN.PICD14/
SADD DAN.SYS-FOROPT
SHDD DAN WRITE4/
SFOR.S DAN.WRITE4/
SADD DAN.SYS-FOROPT
SHDD DAN VER4P/
SFOR.S DAN.VER4P/
SADD DAN.SYS-FOROPT
SHDD DAN VER4Q/
SFOR.S DAN.VER4Q/
SADD DAN.SYS-FOROPT
SHDD DAN VER04U/
SFOR.S DAN.VER04U/
SADD DAN.SYS-FOROPT
SHDD DAN ARGRET/
SASH.FS DAN.ARGRET/
SHDD DAN CLOSE4/
SFOR.S DAN.CLOSE4/
SADD DAN.SYS-FOROPT
SHDD DAN KMDCLE/
SFOR.S DAN.KMDCLE/
SADD DAN.SYS-FOROPT
SHDD DAN KMDAL1/
SFOR.S DAN.KMDAL1/
SADD DAN.SYS-FOROPT
SHDD DAN KMD0EN/
SFOR.S DAN.KMD0EN/
SADD DAN.SYS-FOROPT
SHDD DAN KMDHEA/
SFOR.S DAN.KMDHEA/
SADD DAN.SYS-FOROPT
SHDD DAN KMDNEH/
SFOR.S DAN.KMDNEH/
SADD DAN.SYS-FOROPT
SHDD DAN KMDEXP/
SFOR.S DAN.KMDEXP/
SADD DAN.SYS-FOROPT
SHDD DAN KMDTIM/
SFOR.S DAN.KMDTIM/
SADD DAN.SYS-FOROPT
SHDD DAN KMDSYH/
SFOR.S DAN.KMDSYH/
SADD DAN.SYS-FOROPT
SHDD DAN KMDNER/
SFOR.S DAN.KMDNER/
SADD DAN.SYS-FOROPT
SHDD DAN KMDZON/
SFOR.S DAN.KMDZON/
SADD DAN.SYS-FOROPT

ORIGINAL PAGE IS
OF POOR QUALITY

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
010

SPACK DAM.
SHDG DAM KMDON/
SFOR.S DAM.KMDON/
SADD DAM.SYS-FOROPT
SHDG DAM KMDOFF/
SFOR.S DAM.KMDOFF/
SADD DAM.SYS-FOROPT
SHDG DAM KMDNAM/
SFOR.S DAM.KMDNAM/
SADD DAM.SYS-FOROPT
SHDG DAM KMDORI/
SFOR.S DAM.KMDORI/
SADD DAM.SYS-FOROPT
SHDG DAM CST4IN/
SFOR.S DAM.CST4IN/
SADD DAM.SYS-FOROPT
SHDG DAM CB4IN/
SFOR.S DAM.CB4IN/
SADD DAM.SYS-FOROPT
SHDG DAM KMDCOP/
SFOR.S DAM.KMDCOP/
SADD DAM.SYS-FOROPT
SHDG DAM MDLOG/
SFOR.S DAM.MDLOG/
SADD DAM.SYS-FOROPT
SHDG DAM KMDPOI/
SFOR.S DAM.KMDPOI/
SADD DAM.SYS-FOROPT
SHDG DAM KMDPRI/
SFOR.S DAM.KMDPRI/
SADD DAM.SYS-FOROPT
SHDG DAM KMDXXX/
SFOR.S DAM.KMDXXX/
SADD DAM.SYS-FOROPT
SHDG DAM KMXXED/
SFOR.S DAM.KMXXED/
SADD DAM.SYS-FOROPT
SHDG DAM XREG77/
SASH.FS DAM.XREG77/
SHDG DAM AREG77/
SASH.FS DAM.AREG77/
SHDG DAM KNOLIN/
SFOR.S DAM.KNOLIN/
SADD DAM.SYS-FOROPT
SHDG DAM KMDNEX/
SFOR.S DAM.KMDNEX/
SADD DAM.SYS-FOROPT
SHDG DAM KMDPOL/
SFOR.S DAM.KMDPOL/
SADD DAM.SYS-FOROPT
SHDG DAM KMDCOL/
SFOR.S DAM.KMDCOL/
SADD DAM.SYS-FOROPT
SHDG DAM KOLR4I/
SFOR.S DAM.KOLR4I/
SADD DAM.SYS-FOROPT

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
011

0H00 DAM 14KOLR/
0FOR.S DAM.14KOLR/
0ADD DAM.SYS-FOROPT
0H00 DAM 0CONST/
0FOR.S DAM.0CONST/
0ADD DAM.SYS-FOROPT
0PACK DAM.
0H00 DAM KMDGEO/
0FOR.S DAM.KMDGEO/
0ADD DAM.SYS-FOROPT
0H00 DAM STREG8/
0FOR.S DAM.STREG8/
0ADD DAM.SYS-FOROPT
0H00 DAM KMDATT/
0FOR.S DAM.KMDATT/
0ADD DAM.SYS-FOROPT
0H00 DAM KMDPAG/
0FOR.S DAM.KMDPAG/
0ADD DAM.SYS-FOROPT
0H00 DAM KMDRAD/
0FOR.S DAM.KMDRAD/
0ADD DAM.SYS-FOROPT
0H00 DAM KMDTIC/
0FOR.S DAM.KMDTIC/
0ADD DAM.SYS-FOROPT
0H00 DAM ERTSHP/DAM
0ASH.FS DAM.ERTSHP/DAM
0H00 DAM TRECVR/
0FOR.S DAM.TRECVR/
0ADD DAM.SYS-FOROPT
0H00 DAM PUTHEX/
0FOR.S DAM.PUTHEX/
0ADD DAM.SYS-FOROPT
0H00 DAM CALSPA/
0FOR.S DAM.CALSPA/
0ADD DAM.SYS-FOROPT
0H00 DAM RD3BSQ/
0FOR.S DAM.RD3BSQ/
0ADD DAM.SYS-FOROPT
0H00 DAM ERSPTCN/
0ASH.FS DAM.ERSPTCN/
0H00 DAM ERTSIDC/VIRTUAL
0ASH.FS DAM.ERTSIDC/VIRTUAL
0H00 DAM PICTAB/VIRTUAL
0ASH.FS DAM.PICTAB/VIRTUAL
0H00 DAM CONTROL/VIRTUAL
0ASH.FS DAM.CONTROL/VIRTUAL
0H00 DAM CLASSIFY/VIRTUAL
0ASH.FS DAM.CLASSIFY/VIRTUAL
0H00 DAM PRCLCLASS/VIRTUAL
0ASH.FS DAM.PRCLCLASS/VIRTUAL
0H00 DAM PLTCLASS/VIRTUAL
0ASH.FS DAM.PLTCLASS/VIRTUAL
0H00 DAM STATUS/VIRTUAL
0ASH.FS DAM.STATUS/VIRTUAL
0H00 DAM DITCOP/VIRTUAL

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
012

8ASH.FS DAM.DITCOP/VIRTUAL
8HDD DAM WINEXT/
8FOR.S DAM.WINEXT/
8ADD DAM.SYS-FOROPT
8PACK DAM.
8HDD DAM GETNUL/
8FOR.S DAM.GETNUL/
8ADD DAM.SYS-FOROPT
8HDD DAM CLAEXI/
8FOR.S DAM.CLAEXI/
8ADD DAM.SYS-FOROPT
8HDD DAM RD3NUL/
8FOR.S DAM.RD3NUL/
8ADD DAM.SYS-FOROPT
8HDD DAM OPRPRC/
8FOR.S DAM.OPRPRC/
8ADD DAM.SYS-FOROPT
8HDD DAM CALSYM/
8FOR.S DAM.CALSYM/
8ADD DAM.SYS-FOROPT
8HDD DAM PICDIS/
8FOR.S DAM.PICDIS/
8ADD DAM.SYS-FOROPT
8HDD DAM CLSO2N/
8FOR.S DAM.CLSO2N/
8ADD DAM.SYS-FOROPT
8HDD DAM PICLIS/
8FOR.S DAM.PICLIS/
8ADD DAM.SYS-FOROPT
8HDD DAM PROVSY/
8FOR.S DAM.PROVSY/
8ADD DAM.SYS-FOROPT
8HDD DAM PICP14/
8FOR.S DAM.PICP14/
8ADD DAM.SYS-FOROPT
8HDD DAM PICP15/
8FOR.S DAM.PICP15/
8ADD DAM.SYS-FOROPT
8HDD DAM CB4CST/
8FOR.S DAM.CB4CST/
8ADD DAM.SYS-FOROPT
8HDD DAM CLADE4/
8FOR.S DAM.CLADE4/
8ADD DAM.SYS-FOROPT
8HDD DAM SYSGET/DAM
8ASH.FS DAM.SYSGET/DAM
8HDD DAM ERCSF/DAM
8ASH.FS DAM.ERCSF/DAM
8HDD DAM 8ST488/1108
8ASH.FS DAM.8ST488/1108
8HDD DAM 8ST488/1110
8ASH.FS DAM.8ST488/1110
8HDD DAM 8848ST/1110
8ASH.FS DAM.8848ST/1110
8HDD DAM CB4RL/
8FOR.S DAM.CB4RL/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
013

8ADD DAM.SYS-FOROPT
8HDD DAM CLRQND/
8ASH.FS DAM.CLRQND/
8HDD DAM CST4RL/
8FOR.S DAM.CST4RL/
8ADD DAM.SYS-FOROPT
8PACK DAM.
8HDD DAM CBS4CS/
8FOR.S DAM.CBS4CS/
8ADD DAM.SYS-FOROPT
8HDD DAM CBS4IN/
8FOR.S DAM.CBS4IN/
8ADD DAM.SYS-FOROPT
8HDD DAM CBS4RL/
8FOR.S DAM.CBS4RL/
8ADD DAM.SYS-FOROPT
8HDD DAM GETBYT/
8ASH.FS DAM.GETBYT/
8HDD DAM GETCHR/
8ASH.FS DAM.GETCHR/
8HDD DAM GETDBY/
8ASH.FS DAM.GETDBY/
8HDD DAM TRACE/DAM
8ASH.FS DAM.TRACE/DAM
8HDD DAM GETINT/
8ASH.FS DAM.GETINT/
8HDD DAM GETNYB/
8ASH.FS DAM.GETNYB/
8HDD DAM GETQBY/
8ASH.FS DAM.GETQBY/
8HDD DAM ICE/
8ASH.FS DAM.ICE/
8HDD DAM ICHR/
8ASH.FS DAM.ICHR/
8HDD DAM I4KONE/
8ASH.FS DAM.I4KONE/
8HDD DAM I4KTWO/
8ASH.FS DAM.I4KTWO/
8HDD DAM KONE41/
8ASH.FS DAM.KONE41/
8HDD DAM KTHO41/
8ASH.FS DAM.KTHO41/
8HDD DAM LINTEQ/
8ASH.FS DAM.LINTEQ/
8HDD DAM LINTNE/
8ASH.FS DAM.LINTNE/
8HDD DAM MOV8ST/ASH
8ASH.FS DAM.MOV8ST/ASH
8HDD DAM MOV8YT/
8ASH.FS DAM.MOV8YT/
8HDD DAM MOVCHR/
8ASH.FS DAM.MOVCHR/
8PACK DAM.
8HDD DAM MOV8ST/ASH
8ASH.FS DAM.MOV8ST/ASH
8HDD DAM MOV8BY/
8ASH.FS DAM.MOV8BY/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
014

8ASH.FS DAM.NOVOBY/
8HDO DAM MOVIST/
8ASH.FS DAM.MOVIST/
8HDO DAM NO4NI/
8ASH.FS DAM.NO4NI/
8HDO DAM NC4NI/
8ASH.FS DAM.NC4NI/
8HDO DAM NI4NB/
8ASH.FS DAM.NI4NB/
8HDO DAM NI4NC/
8ASH.FS DAM.NI4NC/
8HDO DAM PUTBYT/
8ASH.FS DAM.PUTBYT/
8HDO DAM PUTCHR/
8ASH.FS DAM.PUTCHR/
8HDO DAM PUTOBY/
8ASH.FS DAM.PUTOBY/
8HDO DAM PUTINT/
8ASH.FS DAM.PUTINT/
8HDO DAM PUTNYB/
8ASH.FS DAM.PUTNYB/
8HDO DAM PUTQBY/
8ASH.FS DAM.PUTQBY/
8HDO DAM SETQWD/
8ASH.FS DAM.SETQWD/
8HDO DAM EB4AS/
8FOR.S DAM.EB4AS/
8ADD DAM.SYS-FOROPT
8HDO DAM AS4CST/
8FOR.S DAM.AS4CST/
8ADD DAM.SYS-FOROPT
8HDO DAM AS4EB/
8FOR.S DAM.AS4EB/
8ADD DAM.SYS-FOROPT
8HDO DAM CST4EB/
8FOR.S DAM.CST4EB/
8ADD DAM.SYS-FOROPT
8HDO DAM OCODE/
8FOR.S DAM.DCODE/
8ADD DAM.SYS-FOROPT
8HDO DAM GETHEX/
8FOR.S DAM.OETHEX/
8ADD DAM.SYS-FOROPT
8HDO DAM LBYTEQ/
8FOR.S DAM.LBYTEQ/
8ADD DAM.SYS-FOROPT
8PACK DAM.
8HDO DAM LBYTNE/
8FOR.S DAM.LBYTNE/
8ADD DAM.SYS-FOROPT
8HDO DAM LCHREQ/
8FOR.S DAM.LCHREQ/
8ADD DAM.SYS-FOROPT
8HDO DAM LCHRNE/
8FOR.S DAM.LCHRNE/
8ADD DAM.SYS-FOROPT

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
015

8H00 DAM LICEEQ/
8FOR.S DAM.LICEEQ/
8ADD DAM.SYS-FOROPT
8H00 DAM LICENE/
8FOR.S DAM.LICENE/
8ADD DAM.SYS-FOROPT
8H00 DAM ATRACE/DAM
8ASM.FS DAM.ATRAE/DAM
8H00 DAM BYTOMP/
8FOR.S DAM.BYTOMP/
8ADD DAM.SYS-FOROPT
8H00 DAM CORLT/
8FOR.S DAM.CORLT/
8ADD DAM.SYS-FOROPT
8H00 DAM DCORLT/
8FOR.S DAM.DCORLT/
8ADD DAM.SYS-FOROPT
8H00 DAM KMDSIZ/
8FOR.S DAM.KMDSIZ/
8ADD DAM.SYS-FOROPT
8H00 DAM KMDCEN/
8FOR.S DAM.KMDCEN/
8ADD DAM.SYS-FOROPT
8H00 DAM KMDSCE/
8FOR.S DAM.KMDSCE/
8ADD DAM.SYS-FOROPT
8H00 DAM PRBNUM/
8FOR.S DAM.PRBNUM/
8ADD DAM.SYS-FOROPT
8H00 DAM KMSPA/
8FOR.S DAM.KMSPA/
8ADD DAM.SYS-FOROPT
8H00 DAM KMDF1/
8FOR.S DAM.KMDF1/
8ADD DAM.SYS-FOROPT
8H00 DAM LBOX41/
8ASM.FS DAM.LBOX41/
8H00 DAM CLA000/
8FOR.S DAM.CLA000/
8ADD DAM.SYS-FOROPT
8H00 DAM CLASSIFY-MAP/
8H00 DAM CST4AS/
8FOR.S DAM.CST4AS/
8ADD DAM.SYS-FOROPT
8H00 DAM READ2N/
8FOR.S DAM.READ2N/
8ADD DAM.SYS-FOROPT
8H00 DAM PRD129/
8FOR.S DAM.PRD129/
8ADD DAM.SYS-FOROPT
8H00 DAM PRD345/
8FOR.S DAM.PRD345/
8ADD DAM.SYS-FOROPT
8PACK DAM.
8H00 DAM PROD19/
8FOR.S DAM.PROD19/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
016

8ADD DAM.SYS-FOROPT
8HDD DAM PRDEXI/
8FOR.S DAM.PRDEXI/
8ADD DAM.SYS-FOROPT
8HDD DAM PROLI9/
8FOR.S DAM.PROLI9/
8ADD DAM.SYS-FOROPT
8HDD DAM PRCHAP/
8FOR.S DAM.PRCHAP/
8ADD DAM.SYS-FOROPT
8HDD DAM PRTDET/VIRTUAL
8ASH.FS DAM.PRTDET/VIRTUAL
8HDD DAM PRTDET-MAP/VIRTUAL
8HDD DAM CONADJ/
8FOR.S DAM.CONADJ/
8ADD DAM.SYS-FOROPT
8HDD DAM CLADE3/
8FOR.S DAM.CLADE3/
8ADD DAM.SYS-FOROPT
8HDD DAM REAL3/
8FOR.S DAM.REAL3/
8ADD DAM.SYS-FOROPT
8HDD DAM RD3BIL/
8FOR.S DAM.RD3BIL/
8ADD DAM.SYS-FOROPT
8HDD DAM R3TREC/
8FOR.S DAM.R3TREC/
8ADD DAM.SYS-FOROPT
8HDD DAM CLOSPR/
8FOR.S DAM.CLOSPR/
8ADD DAM.SYS-FOROPT
8HDD DAM PRSYML/
8FOR.S DAM.PRSYML/
8ADD DAM.SYS-FOROPT
8HDD DAM NITHDD/
8FOR.S DAM.NITHDD/
8ADD DAM.SYS-FOROPT
8HDD DAM MAPHDD/
8FOR.S DAM.MAPHDD/
8ADD DAM.SYS-FOROPT
8HDD DAM KMDPEE/
8FOR.S DAM.KMDPEE/
8ADD DAM.SYS-FOROPT
8HDD DAM KMDPOK/
8FOR.S DAM.KMDPOK/
8ADD DAM.SYS-FOROPT
8HDD DAM EB4CST/
8FOR.S DAM.EB4CST/
8ADD DAM.SYS-FOROPT
8HDD DAM PICPIC/
8FOR.S DAM.PICPIC/
8ADD DAM.SYS-FOROPT
8HDD DAM PROPIC/
8FOR.S DAM.PROPIC/
8ADD DAM.SYS-FOROPT
8HDD DAM DISHIS/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
017

BFOR.S DAM.DISHIS/
BADD DAM.SYS-FOROPT
BHDD DAM CALCOL/
BFOR.S DAM.CALCOL/
BADD DAM.SYS-FOROPT
BPACK DAM.
BHDD DAM PICP19/
BFOR.S DAM.PICP19/
BADD DAM.SYS-FOROPT
BHDD DAM PROPI9/
BFOR.S DAM.PROPI9/
BADD DAM.SYS-FOROPT
BHDD DAM KMDTAB/
BFOR.S DAM.KMDTAB/
BADD DAM.SYS-FOROPT
BHDD DAM KMDIF/
BFOR.S DAM.KMDIF/
BADD DAM.SYS-FOROPT
BHDD DAM RD3DSK/
BFOR.S DAM.RD3DSK/
BADD DAM.SYS-FOROPT
BHDD DAM SETMOD/
BFOR.S DAM.SETMOD/
BADD DAM.SYS-FOROPT
BHDD DAM PITROL/
BFOR.S DAM.PITROL/
BADD DAM.SYS-FOROPT
BHDD DAM LDREGB/
BFOR.S DAM.LDREGB/
BADD DAM.SYS-FOROPT
BHDD DAM PICP13/
BFOR.S DAM.PICP13/
BADD DAM.SYS-FOROPT
BHDD DAM KMDINT/
BFOR.S DAM.KMDINT/
BADD DAM.SYS-FOROPT
BHDD DAM KMDCRO/
BFOR.S DAM.KMDCRO/
BADD DAM.SYS-FOROPT
BHDD DAM PRTOET-MAP/
BHDD DAM PROPI3/
BFOR.S DAM.PROPI3/
BADD DAM.SYS-FOROPT
BHDD DAM IDFILE-MAP/
BHDD DAM IDFILE-MAP/VIRTUAL
BHDD DAM SYPGET/DAM
BASH.FS DAM.SYPGET/DAM
BHDD DAM A40/
BFOR.S DAM.A40/
BADD DAM.SYS-FOROPT
BHDD DAM 04A/
BFOR.S DAM.04A/
BADD DAM.SYS-FOROPT
BHDD DAM 04P/
BFOR.S DAM.04P/
BADD DAM.SYS-FOROPT

ORIGINAL PAGE IS
OF POOR QUALITY

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
010

8H00 DAM 04U/CLARKE1868
8FOR.S DAM.04U/CLARKE1868
8ADD DAM.SYS-FOROPT
8H00 DAM P4A/
8FOR.S DAM.P4A/
8ADD DAM.SYS-FOROPT
8H00 DAM A4P/
8FOR.S DAM.A4P/
8ADD DAM.SYS-FOROPT
8H00 DAM P40/
8FOR.S DAM.P40/
8ADD DAM.SYS-FOROPT
8H00 DAM DU40/CLARKE1868
8FOR.S DAM.DU40/CLARKE1868
8ADD DAM.SYS-FOROPT
8PACK DAM.
8H00 DAM U40/CLARKE1868
8FOR.S DAM.U40/CLARKE1868
8ADD DAM.SYS-FOROPT
8H00 DAM PICD13/
8FOR.S DAM.PICD13/
8ADD DAM.SYS-FOROPT
8H00 DAM PICTAB-MAP/
8H00 DAM PICEX1/
8FOR.S DAM.PICEX1/
8ADD DAM.SYS-FOROPT
8H00 DAM PICFA3/
8FOR.S DAM.PICFA3/
8ADD DAM.SYS-FOROPT
8H00 DAM PICFAC/
8FOR.S DAM.PICFAC/
8ADD DAM.SYS-FOROPT
8H00 DAM PICROT/
8FOR.S DAM.PICROT/
8ADD DAM.SYS-FOROPT
8H00 DAM CONTROL-MAP/
8H00 DAM CON000/
8FOR.S DAM.CON000/
8ADD DAM.SYS-FOROPT
8H00 DAM OPNO2N/
8FOR.S DAM.OPNO2N/
8ADD DAM.SYS-FOROPT
8H00 DAM PRD000/
8FOR.S DAM.PRD000/
8ADD DAM.SYS-FOROPT
8H00 DAM PROL1S/
8FOR.S DAM.PROL1S/
8ADD DAM.SYS-FOROPT
8H00 DAM PROL13/
8FOR.S DAM.PROL13/
8ADD DAM.SYS-FOROPT
8H00 DAM IDCPRD/
8FOR.S DAM.IDCPRD/
8ADD DAM.SYS-FOROPT
8H00 DAM PRC000/
8FOR.S DAM.PRC000/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
010

3ADD DAM.SYS-FOROPT
3HDD DAM CLSHDD/
3FOR.S DAM.CLSHDD/
3ADD DAM.SYS-FOROPT
3HDD DAM OPEN3/
3FOR.S DAM.OPEN3/
3ADD DAM.SYS-FOROPT
3HDD DAM OPN12N/
3FOR.S DAM.OPN12N/
3ADD DAM.SYS-FOROPT
3HDD DAM OP3DSK/
3FOR.S DAM.OP3DSK/
3ADD DAM.SYS-FOROPT
3HDD DAM PXSDMP/
3FOR.S DAM.PXSDMP/
3ADD DAM.SYS-FOROPT
3HDD DAM RD3BIP/
3FOR.S DAM.RD3BIP/
3ADD DAM.SYS-FOROPT
3HDD DAM KMDCHA/
3FOR.S DAM.KMDCHA/
3ADD DAM.SYS-FOROPT
3HDD DAM DETCHA/
3FOR.S DAM.DETCHA/
3ADD DAM.SYS-FOROPT
3PACK DAM.
3HDD DAM CALWIN/
3FOR.S DAM.CALWIN/
3ADD DAM.SYS-FOROPT
3HDD DAM CLOSE3/
3FOR.S DAM.CLOSE3/
3ADD DAM.SYS-FOROPT
3HDD DAM CL3BIP/
3FOR.S DAM.CL3BIP/
3ADD DAM.SYS-FOROPT
3HDD DAM CROPOW/
3FOR.S DAM.CROPOW/
3ADD DAM.SYS-FOROPT
3HDD DAM DSSPR/
3FOR.S DAM.DSSPR/
3ADD DAM.SYS-FOROPT
3HDD DAM GENTIC/
3FOR.S DAM.GENTIC/
3ADD DAM.SYS-FOROPT
3HDD DAM GETDSR/
3FOR.S DAM.GETDSR/
3ADD DAM.SYS-FOROPT
3HDD DAM IDLU3/
3FOR.S DAM.IDLU3/
3ADD DAM.SYS-FOROPT
3HDD DAM ISRTMS/
3FOR.S DAM.ISRTMS/
3ADD DAM.SYS-FOROPT
3HDD DAM ISRTSA/
3FOR.S DAM.ISRTSA/
3ADD DAM.SYS-FOROPT

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
020

SHDD DAM ISRTMA/
SFOR.S DAM.ISRTMA/
SADD DAM.SYS-FOROPT
SHDD DAM ISRTMD/
SFOR.S DAM.ISRTMD/
SADD DAM.SYS-FOROPT
SHDD DAM ISRTSD/
SFOR.S DAM.ISRTSD/
SADD DAM.SYS-FOROPT
SHDD DAM MATPRT/
SFOR.S DAM.MATPRT/
SADD DAM.SYS-FOROPT
SHDD DAM MVCONT/
SASH.FS DAM.MVCONT/
SHDD DAM NVIATO/
SASH.FS DAM.NVIATO/
SHDD DAM OPENPR/
SFOR.S DAM.OPENPR/
SADD DAM.SYS-FOROPT
SHDD DAM OP3TAP/
SFOR.S DAM.OP3TAP/
SADD DAM.SYS-FOROPT
SHDD DAM O3ANCL/
SFOR.S DAM.O3ANCL/
SADD DAM.SYS-FOROPT
SHDD DAM O3ANOT/
SFOR.S DAM.O3ANOT/
SADD DAM.SYS-FOROPT
SHDD DAM O3HDR/
SFOR.S DAM.O3HDR/
SADD DAM.SYS-FOROPT
SPACK DAM.
SHDD DAM O3SZAR/
SFOR.S DAM.O3SZAR/
SADD DAM.SYS-FOROPT
SHDD DAM O3SZAM/
SFOR.S DAM.O3SZAM/
SADD DAM.SYS-FOROPT
SHDD DAM O3SZPM/
SFOR.S DAM.O3SZPM/
SADD DAM.SYS-FOROPT
SHDD DAM O3SZPR/
SFOR.S DAM.O3SZPR/
SADD DAM.SYS-FOROPT
SHDD DAM O3TOR/
SFOR.S DAM.O3TOR/
SADD DAM.SYS-FOROPT
SHDD DAM CALCHA/
SFOR.S DAM.CALCHA/
SADD DAM.SYS-FOROPT
SHDD DAM KMDREN/
SFOR.S DAM.KMDREN/
SADD DAM.SYS-FOROPT
SHDD DAM KMDWIN/
SFOR.S DAM.KMDWIN/
SADD DAM.SYS-FOROPT

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
021

SHDD DAM KNDSCA/
SFOR.S DAM.KNDSCA/
SADD DAM.SYS-FOROPT
SHDD DAM KNDSHA/
SFOR.S DAM.KNDSHA/
SADD DAM.SYS-FOROPT
SHDD DAM KNDTOL/
SFOR.S DAM.KNDTOL/
SADD DAM.SYS-FOROPT
SHDD DAM ERINFO/DAM
SASH.FS DAM.ERINFO/DAM
SHDD DAM USHAP/
SFOR.S DAM.USHAP/
SADD DAM.SYS-FOROPT
SHDD DAM ERTSIDC/
SFOR.S DAM.ERTSIDC/
SADD DAM.SYS-FOROPT
SHDD DAM PICTAB/
SFOR.S DAM.PICTAB/
SADD DAM.SYS-FOROPT
SHDD DAM PIC000/
SFOR.S DAM.PIC000/
SADD DAM.SYS-FOROPT
SHDD DAM PICFAS/
SFOR.S DAM.PICFAS/
SADD DAM.SYS-FOROPT
SHDD DAM PICL13/
SFOR.S DAM.PICL13/
SADD DAM.SYS-FOROPT
SHDD DAM PICKQT/
SFOR.S DAM.PICKQT/
SADD DAM.SYS-FOROPT
SHDD DAM IDC PIC/
SFOR.S DAM.IDCPIC/
SADD DAM.SYS-FOROPT
SHDD DAM OPRPIC/
SFOR.S DAM.OPRPIC/
SADD DAM.SYS-FOROPT
SPACK DAM.
SHDD DAM CONTROL/
SFOR.S DAM.CONTROL/
SADD DAM.SYS-FOROPT
SHDD DAM CONXQT/
SFOR.S DAM.CONXQT/
SADD DAM.SYS-FOROPT
SHDD DAM HSKPIX/
SFOR.S DAM.HSKPIX/
SADD DAM.SYS-FOROPT
SHDD DAM CLAXQT/
SFOR.S DAM.CLAXQT/
SADD DAM.SYS-FOROPT
SHDD DAM PRTOET/
SFOR.S DAM.PRTOET/
SADD DAM.SYS-FOROPT
SHDD DAM CLASSIFY/
SFOR.S DAM.CLASSIFY/

ORIGINAL PAGE IS
OF POOR QUALITY

DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
022

SADD DAN.SYS-FOROPT
SMDD DAN PRODIS/
SFOR.S DAN.PRODIS/
SADD DAN.SYS-FOROPT
SMDD DAN PRODIZ/
SFOR.S DAN.PRODIZ/
SADD DAN.SYS-FOROPT
SMDD DAN PRDXQT/
SFOR.S DAN.PRDXQT/
SADD DAN.SYS-FOROPT
SMDD DAN OPRPRD/
SFOR.S DAN.OPRPRD/
SADD DAN.SYS-FOROPT
SMDD DAN PRTCLASS/
SFOR.S DAN.PRTCLASS/
SADD DAN.SYS-FOROPT
SMDD DAN PRCXQT/
SFOR.S DAN.PRCXQT/
SADD DAN.SYS-FOROPT
SMDD DAN RESPRC/
SFOR.S DAN.RESPRC/
SADD DAN.SYS-FOROPT
SMDD DAN PLC000/
SFOR.S DAN.PLC000/
SADD DAN.SYS-FOROPT
SMDD DAN DIT000/
SFOR.S DAN.DIT000/
SADD DAN.SYS-FOROPT
SMDD DAN DITDUP/
SFOR.S DAN.DITDUP/
SADD DAN.SYS-FOROPT
SMDD DAN DOPCNT/
SFOR.S DAN.DOPCNT/
SADD DAN.SYS-FOROPT
SMDD DAN DORECP/
SFOR.S DAN.DORECP/
SADD DAN.SYS-FOROPT
SMDD DAN DOSQRT/
SFOR.S DAN.DOSQRT/
SADD DAN.SYS-FOROPT
SMDD DAN DMPHIN/
SFOR.S DAN.DMPHIN/
SADD DAN.SYS-FOROPT
SMDD DAN DSSPR3/
SFOR.S DAN.DSSPR3/
SADD DAN.SYS-FOROPT
SPACK DAN.
SMDD DAN EISRTD/
SFOR.S DAN.EISRTD/
SADD DAN.SYS-FOROPT
SMDD DAN ENVORI/
SFOR.S DAN.ENVORI/
SADD DAN.SYS-FOROPT
SMDD DAN ENVWIN/
SFOR.S DAN.ENVWIN/
SADD DAN.SYS-FOROPT

DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
023

SHDS DAN FACPRY/
SFOR.S DAN.FACPRY/
SADD DAN.SYS-FOROPT
SHDS DAN FLINFO/
SFOR.S DAN.FLINFO/
SADD DAN.SYS-FOROPT
SHDS DAN GCERT/
SFOR.S DAN.GCERT/
SADD DAN.SYS-FOROPT
SHDS DAN OCHOM/
SFOR.S DAN.OCHOM/
SADD DAN.SYS-FOROPT
SHDS DAN GETRAD/
SFOR.S DAN.GETRAD/
SADD DAN.SYS-FOROPT
SHDS DAN HDUNIT/
SFOR.S DAN.HDUNIT/
SADD DAN.SYS-FOROPT
SHDS DAN IDERTS/
SFOR.S DAN.IDERTS/
SADD DAN.SYS-FOROPT
SHDS DAN JOINZN/
SFOR.S DAN.JOINZN/
SADD DAN.SYS-FOROPT
SHDS DAN OP3BIP/
SFOR.S DAN.OP3BIP/
SADD DAN.SYS-FOROPT
SHDS DAN PROVFI/
SFOR.S DAN.PROVFI/
SADD DAN.SYS-FOROPT
SHDS DAN PRITCHR/
SFOR.S DAN.PRITCHR/
SADD DAN.SYS-FOROPT
SHDS DAN PX4AH/
SFOR.S DAN.PX4AH/
SADD DAN.SYS-FOROPT
SHDS DAN PX4AR/
SFOR.S DAN.PX4AR/
SADD DAN.SYS-FOROPT
SHDS DAN PX4PH/
SFOR.S DAN.PX4PH/
SADD DAN.SYS-FOROPT
SHDS DAN PX4PR/
SFOR.S DAN.PX4PR/
SADD DAN.SYS-FOROPT
SHDS DAN SHASAH/
SFOR.S DAN.SHASAH/
SADD DAN.SYS-FOROPT
SHDS DAN TSHAP3/
SFOR.S DAN.TSHAP3/
SADD DAN.SYS-FOROPT
SHDS DAN PRTRG/
SFOR.S DAN.PRTRG/
SADD DAN.SYS-FOROPT
SPACK DAN.
SHDS DAN PRTINC/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE
024

3FOR.S DAM.PRTINC/
3ADD DAM.SYS-FOROPT
3HOG DAM DEG/
3FOR.S DAM.DEG/
3ADD DAM.SYS-FOROPT
3HOG DAM D2OMS/
3FOR.S DAM.D2OMS/
3ADD DAM.SYS-FOROPT
3HOG DAM SPLIT/
3FOR.S DAM.SPLIT/
3ADD DAM.SYS-FOROPT
3HOG DAM CLADET/
3FOR.S DAM.CLADET/
3ADD DAM.SYS-FOROPT
3HOG DAM MAPRNT/
3FOR.S DAM.MAPRNT/
3ADD DAM.SYS-FOROPT
3HOG DAM IDFILE/
3FOR.S DAM.IDFILE/
3ADD DAM.SYS-FOROPT
3HOG DAM LOCATE/
3FOR.S DAM.LOCATE/
3ADD DAM.SYS-FOROPT
3HOG DAM OP3MOP/
3FOR.S DAM.OP3MOP/
3ADD DAM.SYS-FOROPT
3HOG DAM CONDIA/
3FOR.S DAM.CONDIA/
3ADD DAM.SYS-FOROPT
3HOG DAM PRCLASS-MAP/
3HOG DAM STATUS/
3FOR.S DAM.STATUS/
3ADD DAM.SYS-FOROPT

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COLLECT
001

```
0MS0.N  SYS-COLLECT RUNSTREAM FOR 0MAP PROCESSOR
0MS0.N  -----
0MS0.N
0MS0.N
0MS0.N  HISTORY
0MS0.N  -----
0MS0.N
0MS0.N  E H SCHLOSSER      LEC      03/03/73      ORIGINAL CODE
0MS0.N  E H SCHLOSSER      LEC      07/31/79      ADD F OPTION (QUARTER-WORD MODE)
0MS0.N  E H SCHLOSSER      LEMSCO   05/16/80      CHANGE PRDENS TO PRDDET
0MS0.N
0MS0.N  EXCEPTIONS
0MS0.N  -----
0MS0.N
0MS0.N  1. THIS ELEMENT MUST NOT BE 0ADD-ED FROM THE DAM PROGRAM FILE.
0MS0.N  SINCE IT 0PACK-S THAT FILE. INSTEAD IT MUST BE COPIED TO
0MS0.N  TPFS. AND 0ADD-ED FROM THERE.
0MS0.N
0MS0.N
0MDO DAM PACKAGE -- COMPILE SYS-BLOCK AND COLLECT MAIN PROGRAMS
0FOR.S DAM.SYS-BLOCK
0ADD DAM.SYS-FOROPT
0PACK.SR DAM. . CAUTION: ADD ELEMENT MUST NOT PACK ITS OWN FILE!
0PREP DAM.
0MAP.FLZ DAM.ERSPRTCN-MAP.DAM.ERSPRTCN . NEVER VIRTUAL!!
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.IDFILE-MAP.DAM.IDFILE . NEVER VIRTUAL!!
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.LOCATE-MAP.DAM.LOCATE . NEVER VIRTUAL!!
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.USWAP-MAP.DAM.USWAP . NEVER VIRTUAL!!
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.ERTSIDC-MAP/VIRTUAL.DAM.ERTSIDC
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.PICTAB-MAP.DAM.PICTAB
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.CONTROL-MAP.DAM.CONTROL
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.CLASSIFY-MAP/VIRTUAL.DAM.CLASSIFY
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.PRDDET-MAP/VIRTUAL.DAM.PRDDET
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.PRTCLASS-MAP/VIRTUAL.DAM.PRTCLASS
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.PLTCLASS-MAP/VIRTUAL.DAM.PLTCLASS
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.STATUS-MAP.DAM.STATUS
0ADD DAM.SYS-MAPOPT
0MAP.FLZ DAM.OITCOP-MAP/VIRTUAL.DAM.OITCOP
0ADD DAM.SYS-MAPOPT
```

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
001

0DELETE.S DAM.ERPRCN/DAM
0DELETE.S DAM.DLSTSQ/
0DELETE.S DAM.DIAERP/
0DELETE.S DAM.GAPFOC/
0DELETE.S DAM.GAPLUF/
0DELETE.S DAM.GAPLUR/
0DELETE.S DAM.FOCUS/
0DELETE.S DAM.NULSUB/
0DELETE.S DAM.DMPTIC/
0DELETE.S DAM.PICPR3/
0DELETE.S DAM.ERERR/DAM
0DELETE.S DAM.EREXIT/DAM
0DELETE.S DAM.PICD15/
0DELETE.S DAM.PICPA6/
0DELETE.S DAM.184350/
0DELETE.S DAM.PICL15/
0DELETE.S DAM.PICPA9/
0DELETE.S DAM.PICPR9/
0DELETE.S DAM.PICT03/
0DELETE.S DAM.PICT09/
0DELETE.S DAM.CLADES/
0DELETE.S DAM.CLADE9/
0DELETE.S DAM.OPRCLA/
0DELETE.S DAM.SLMCLA/
0DELETE.S DAM.COPYPX/
0DELETE.S DAM.PIC129/
0DELETE.S DAM.PIC345/
0DELETE.S DAM.PIC678/
0DELETE.S DAM.CLA129/
0DELETE.S DAM.CLA345/
0DELETE.S DAM.H2T0H1/
0DELETE.S DAM.HILL0L/
0DELETE.S DAM.IDERT/
0DELETE.S DAM.INVORI/
0DELETE.S DAM.INVWIN/
0DELETE.S DAM.ISRTBA/
0DELETE.S DAM.ISRTBD/
0DELETE.S DAM.KSPRED/
0DELETE.S DAM.LSPRED/
0DELETE.S DAM.MSPRED/
0DELETE.S DAM.NEGPIC/
0DELETE.S DAM.PRM1PX/
0DELETE.S DAM.PRM2PX/
0DELETE.S DAM.QUAD/
0DELETE.S DAM.RITADD/
0DELETE.S DAM.RL21SX/
0DELETE.S DAM.RL2SX/
0DELETE.S DAM.RL4SX/
0DELETE.S DAM.ROTCHX/
0DELETE.S DAM.ROTCOL/
0DELETE.S DAM.ROTR0W/
0DELETE.S DAM.SHFTBC/
0DELETE.S DAM.SSPR/
0DELETE.S DAM.SUBH1/
0DELETE.S DAM.VALKEY/
0DELETE.S DAM.VALL0L/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
002

0DELETE.S DAM.HRVERT/
0DELETE.S DAM.EAPRNT/DAM
0DELETE.S DAM.EAREAD/DAM
0DELETE.S DAM.ERDATE/DAM
0DELETE.S DAM.ERFACL/DAM
0DELETE.S DAM.ERFITH/DAM
0DELETE.S DAM.ERPCHA/DAM
0DELETE.S DAM.ERPCT/DAM
0DELETE.S DAM.ERPRCA/DAM
0DELETE.S DAM.ERPRNT/DAM
0DELETE.S DAM.ERPRTA/DAM
0DELETE.S DAM.ERREAD/DAM
0DELETE.S DAM.ERREDA/DAM
0DELETE.S DAM.ERSUPS/DAM
0DELETE.S DAM.ERTHAT/DAM
0DELETE.S DAM.PICPRO/
0DELETE.S DAM.PICTOT/
0DELETE.S DAM.KMDOAS/
0DELETE.S DAM.KMDOFR/
0DELETE.S DAM.KHR4IN/
0DELETE.S DAM.ERPFS/DAM
0DELETE.S DAM.DITVER/
0DELETE.S DAM.REVERT/
0DELETE.S DAM.LOCDSF/
0DELETE.S DAM.KMDOAD/
0DELETE.S DAM.KMDOLO/
0DELETE.S DAM.LENPAD/
0DELETE.S DAM.LOWCST/
0DELETE.S DAM.SREADS/
0DELETE.S DAM.SPANS/
0DELETE.S DAM.GETOKH/
0DELETE.S DAM.NEXTOK/
0DELETE.S DAM.GETS/
0DELETE.S DAM.ERIO/DAM
0DELETE.S DAM.ERLOW/DAM
0DELETE.S DAM.ERWAIT/DAM
0DELETE.S DAM.TRUAL/
0DELETE.S DAM.TRUCST/
0DELETE.S DAM.KMDOBR/
0DELETE.S DAM.QUARTN/
0DELETE.S DAM.QUARTU/
0DELETE.S DAM.VARSQU/
0DELETE.S DAM.VARSON/
0DELETE.S DAM.SYSADD/DAM
0DELETE.S DAM.KMXXGS/
0DELETE.S DAM.CALSCA/
0DELETE.S DAM.KMDCOU/
0DELETE.S DAM.DITXQT/
0DELETE.S DAM.SUBWIN/
0DELETE.S DAM.PRCXI/
0DELETE.S DAM.PLTCLASS/
0DELETE.S DAM.DITCOP/
0DELETE.S DAM.DITEXI/
0DELETE.S DAM.CONEXI/
0DELETE.S DAM.PLCXQT/
0DELETE.S DAM.KMDPLO/

ORIGINAL PAGE IS
OF POOR QUALITY

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
003

0DELETE.S DAM.LCSTEQ/
0DELETE.S DAM.PLCHAP/
0DELETE.S DAM.PSTOP/
0DELETE.S DAM.CLARAD/
0DELETE.S DAM.PICPA4/
0DELETE.S DAM.PICPA3/
0DELETE.S DAM.PICPAR/
0DELETE.S DAM.PICLI4/
0DELETE.S DAM.PICDI9/
0DELETE.S DAM.PICLI9/
0DELETE.S DAM.LOG2/
0DELETE.S DAM.NTAB5/DAM
0DELETE.S DAM.CBINIT/
0DELETE.S DAM.WARN5/
0DELETE.S DAM.IDUP/
0DELETE.S DAM.LENCST/
0DELETE.S DAM.PSTART/
0DELETE.S DAM.READ5/
0DELETE.S DAM.PICDI4/
0DELETE.S DAM.WRITE4/
0DELETE.S DAM.VERA4P/
0DELETE.S DAM.VERA4O/
0DELETE.S DAM.VERO4U/
0DELETE.S DAM.ARGRET/
0DELETE.S DAM.CLOSE4/
0DELETE.S DAM.KMOCLE/
0DELETE.S DAM.KMOALI/
0DELETE.S DAM.KMDDEN/
0DELETE.S DAM.KMONEA/
0DELETE.S DAM.KMONEW/
0DELETE.S DAM.KMDEXP/
0DELETE.S DAM.KMOTIM/
0DELETE.S DAM.KMOSYM/
0DELETE.S DAM.KMOMER/
0DELETE.S DAM.KMDOZON/
0DELETE.S DAM.KMDOFF/
0DELETE.S DAM.KMDNAM/
0DELETE.S DAM.KMDORI/
0DELETE.S DAM.CST4IN/
0DELETE.S DAM.CB4IN/
0DELETE.S DAM.KMDCOP/
0DELETE.S DAM.MDLOG/
0DELETE.S DAM.KMDPOI/
0DELETE.S DAM.KMDPRI/
0DELETE.S DAM.KMDXXX/
0DELETE.S DAM.KMXXED/
0DELETE.S DAM.XRE077/
0DELETE.S DAM.ARE077/
0DELETE.S DAM.KMOLIN/
0DELETE.S DAM.KMDNEX/
0DELETE.S DAM.KMDPOL/
0DELETE.S DAM.KMDCOL/
0DELETE.S DAM.KOLR4I/
0DELETE.S DAM.I4KOLR/
0DELETE.S DAM.OCONST/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
004

DELETE.S DAM.KM0GEO/
DELETE.S DAM.STREG0/
DELETE.S DAM.KMDATT/
DELETE.S DAM.KMDPAG/
DELETE.S DAM.KMDRAD/
DELETE.S DAM.KMDTIC/
DELETE.S DAM.ERTSWP/DAM
DELETE.S DAM.TRECVR/
DELETE.S DAM.PUTHEX/
DELETE.S DAM.CALSPA/
DELETE.S DAM.RD3BSQ/
DELETE.S DAM.ERSPRTCN/
DELETE.S DAM.ERTSIDC/VIRTUAL
DELETE.S DAM.PICTAB/VIRTUAL
DELETE.S DAM.CONTROL/VIRTUAL
DELETE.S DAM.CLASSIFY/VIRTUAL
DELETE.S DAM.PRTCLASS/VIRTUAL
DELETE.S DAM.PLTCLASS/VIRTUAL
DELETE.S DAM.STATUS/VIRTUAL
DELETE.S DAM.DITCOP/VIRTUAL
DELETE.S DAM.WINEXT/
DELETE.S DAM.GETNUL/
DELETE.S DAM.CLAEXI/
DELETE.S DAM.RD3NUL/
DELETE.S DAM.OPRPRC/
DELETE.S DAM.CALSYM/
DELETE.S DAM.PICDIS/
DELETE.S DAM.CLSO2N/
DELETE.S DAM.PICLIS/
DELETE.S DAM.PROVSY/
DELETE.S DAM.PICP14/
DELETE.S DAM.PICP15/
DELETE.S DAM.CB4CST/
DELETE.S DAM.CLADE4/
DELETE.S DAM.SYSOET/DAM
DELETE.S DAM.ERCSF/DAM
DELETE.S DAM.BST488/1108
DELETE.S DAM.BST488/1110
DELETE.S DAM.B84BST/1110
DELETE.S DAM.CB4RL/
DELETE.S DAM.CLQWD/
DELETE.S DAM.CST4RL/
DELETE.S DAM.CB54CS/
DELETE.S DAM.CB54IN/
DELETE.S DAM.CB54RL/
DELETE.S DAM.GETBYT/
DELETE.S DAM.GETCHR/
DELETE.S DAM.GETOBY/
DELETE.S DAM.TRACE/DAM
DELETE.S DAM.GETINT/
DELETE.S DAM.GETNYB/
DELETE.S DAM.GETOBY/
DELETE.S DAM.ICE/
DELETE.S DAM.ICHR/
DELETE.S DAM.I4KONE/
DELETE.S DAM.I4KTWO/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
005

0DELETE.S DAM.KONE41/
0DELETE.S DAM.KTHO41/
0DELETE.S DAM.LINTEQ/
0DELETE.S DAM.LINTNE/
0DELETE.S DAM.MOV8ST/ASM
0DELETE.S DAM.MOV8YT/
0DELETE.S DAM.MOVCHR/
0DELETE.S DAM.MOVCSST/ASM
0DELETE.S DAM.MOV8BY/
0DELETE.S DAM.MOVIST/
0DELETE.S DAM.NB4NI/
0DELETE.S DAM.NC4NI/
0DELETE.S DAM.NI4NB/
0DELETE.S DAM.NI4NC/
0DELETE.S DAM.PUT8YT/
0DELETE.S DAM.PUTCHR/
0DELETE.S DAM.PUT8BY/
0DELETE.S DAM.PUTINT/
0DELETE.S DAM.PUTNY8/
0DELETE.S DAM.PUT8BY/
0DELETE.S DAM.SETQND/
0DELETE.S DAM.E84AS/
0DELETE.S DAM.AS4CST/
0DELETE.S DAM.AS4EB/
0DELETE.S DAM.CST4EB/
0DELETE.S DAM.DCODE/
0DELETE.S DAM.0ETHEX/
0DELETE.S DAM.LBYTEQ/
0DELETE.S DAM.LBYTNE/
0DELETE.S DAM.LCHREQ/
0DELETE.S DAM.LCHRNE/
0DELETE.S DAM.LICEEQ/
0DELETE.S DAM.LICENE/
0DELETE.S DAM.ATRACE/DAM
0DELETE.S DAM.BYTDMP/
0DELETE.S DAM.CORLT/
0DELETE.S DAM.DCORLT/
0DELETE.S DAM.KMDSIZ/
0DELETE.S DAM.KMDCEN/
0DELETE.S DAM.KMDSCE/
0DELETE.S DAM.PRBNUM/
0DELETE.S DAM.KMDSPA/
0DELETE.S DAM.KMDFI/
0DELETE.S DAM.LBOX41/
0DELETE.S DAM.CLA000/
0DELETE.S DAM.CST4AS/
0DELETE.S DAM.READ2N/
0DELETE.S DAM.PRO129/
0DELETE.S DAM.PRO345/
0DELETE.S DAM.PROD19/
0DELETE.S DAM.PROEX1/
0DELETE.S DAM.PROL19/
0DELETE.S DAM.PRCHAP/
0DELETE.S DAM.PRTOET/VIRTUAL
0DELETE.S DAM.CONADJ/
0DELETE.S DAM.CLADE3/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
006

8DELETE.S DAM.READ3/
8DELETE.S DAM.RD3BIL/
8DELETE.S DAM.R3TREC/
8DELETE.S DAM.CLOSPR/
8DELETE.S DAM.PRSYML/
8DELETE.S DAM.NITHOG/
8DELETE.S DAM.MAPHOG/
8DELETE.S DAM.KHOPEE/
8DELETE.S DAM.KHOPOK/
8DELETE.S DAM.EB4CST/
8DELETE.S DAM.PICPIC/
8DELETE.S DAM.PROPIC/
8DELETE.S DAM.DISHIS/
8DELETE.S DAM.CALCOL/
8DELETE.S DAM.PICPI9/
8DELETE.S DAM.PROPI9/
8DELETE.S DAM.KHOTAB/
8DELETE.S DAM.KHOIF/
8DELETE.S DAM.RD3DSK/
8DELETE.S DAM.SETMOD/
8DELETE.S DAM.PITROL/
8DELETE.S DAM.LDREG8/
8DELETE.S DAM.PICPI3/
8DELETE.S DAM.KHOINT/
8DELETE.S DAM.KMOCRO/
8DELETE.S DAM.PROPI3/
8DELETE.S DAM.SYPOET/DAM
8DELETE.S DAM.A4G/
8DELETE.S DAM.G4A/
8DELETE.S DAM.G4P/
8DELETE.S DAM.G4U/CLARKE1866
8DELETE.S DAM.P4A/
8DELETE.S DAM.A4P/
8DELETE.S DAM.P4G/
8DELETE.S DAM.DU4G/CLARKE1866
8DELETE.S DAM.U4G/CLARKE1866
8DELETE.S DAM.PICD13/
8DELETE.S DAM.PICEX1/
8DELETE.S DAM.PICFA3/
8DELETE.S DAM.PICFAC/
8DELETE.S DAM.PICROT/
8DELETE.S DAM.CON000/
8DELETE.S DAM.OPN02N/
8DELETE.S DAM.PR0000/
8DELETE.S DAM.PROL15/
8DELETE.S DAM.PROL13/
8DELETE.S DAM.IDCPRO/
8DELETE.S DAM.PRC000/
8DELETE.S DAM.CLSH00/
8DELETE.S DAM.OPEN3/
8DELETE.S DAM.OPN12N/
8DELETE.S DAM.OP30SK/
8DELETE.S DAM.PX80MP/
8DELETE.S DAM.RD3BIP/
8DELETE.S DAM.KMOCHA/
8DELETE.S DAM.DETCHA/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
007

8DELETE.S DAM.CALWIN/
8DELETE.S DAM.CLOSE3/
8DELETE.S DAM.CL38IP/
8DELETE.S DAM.CROPOH/
8DELETE.S DAM.DSSPR/
8DELETE.S DAM.GENTIC/
8DELETE.S DAM.GETOSR/
8DELETE.S DAM.IDLU3/
8DELETE.S DAM.TSRTMS/
8DELETE.S DAM.ISRTSA/
8DELETE.S DAM.ISRTMA/
8DELETE.S DAM.ISRTHD/
8DELETE.S DAM.ISRTSD/
8DELETE.S DAM.MATPRT/
8DELETE.S DAM.MVCONT/
8DELETE.S DAM.NVIATO/
8DELETE.S DAM.OPENPR/
8DELETE.S DAM.OP3TAP/
8DELETE.S DAM.O3ANCL/
8DELETE.S DAM.O3ANOT/
8DELETE.S DAM.O3HDR/
8DELETE.S DAM.O3SZAR/
8DELETE.S DAM.O3SZAM/
8DELETE.S DAM.O3SZPM/
8DELETE.S DAM.O3SZPR/
8DELETE.S DAM.O3TDR/
8DELETE.S DAM.CALCHA/
8DELETE.S DAM.KHOREN/
8DELETE.S DAM.KMDWIN/
8DELETE.S DAM.KMOSCA/
8DELETE.S DAM.KMOSHA/
8DELETE.S DAM.KMDTOL/
8DELETE.S DAM.ERINFO/DAM
8DELETE.S DAM.USWAP/
8DELETE.S DAM.ERTSIDC/
8DELETE.S DAM.PICTAB/
8DELETE.S DAM.PIC000/
8DELETE.S DAM.PICFA9/
8DELETE.S DAM.PICL13/
8DELETE.S DAM.PICXQT/
8DELETE.S DAM.IDCPIC/
8DELETE.S DAM.OPRPIC/
8DELETE.S DAM.CONTROL/
8DELETE.S DAM.CONXQT/
8DELETE.S DAM.MSKPIX/
8DELETE.S DAM.CLAXQT/
8DELETE.S DAM.PRTDET/
8DELETE.S DAM.CLASSIFY/
8DELETE.S DAM.PROD1S/
8DELETE.S DAM.PROD13/
8DELETE.S DAM.PROXQT/
8DELETE.S DAM.OPRPRO/
8DELETE.S DAM.PRTCLASS/
8DELETE.S DAM.PRCXQT/
8DELETE.S DAM.RESPRC/
8DELETE.S DAM.PLC000/

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
000

8DELETE.S DAM.DIT000/
8DELETE.S DAM.DITDUP/
8DELETE.S DAM.DOPCNT/
8DELETE.S DAM.DGRECP/
8DELETE.S DAM.DOSQRT/
8DELETE.S DAM.DMWIN/
8DELETE.S DAM.DSSPR3/
8DELETE.S DAM.EISRTD/
8DELETE.S DAM.ENVORI/
8DELETE.S DAM.ENVWIN/
8DELETE.S DAM.FACPR1/
8DELETE.S DAM.FLINFD/
8DELETE.S DAM.GCERT/
8DELETE.S DAM.GCHOM/
8DELETE.S DAM.GETRAD/
8DELETE.S DAM.HDUNIT/
8DELETE.S DAM.IDERTS/
8DELETE.S DAM.JOIN2N/
8DELETE.S DAM.OP3BIP/
8DELETE.S DAM.PROVFI/
8DELETE.S DAM.PRTCHR/
8DELETE.S DAM.PX4AM/
8DELETE.S DAM.PX4AR/
8DELETE.S DAM.PX4PM/
8DELETE.S DAM.PX4PR/
8DELETE.S DAM.SHASAM/
8DELETE.S DAM.TSHAP3/
8DELETE.S DAM.PRTMRO/
8DELETE.S DAM.PRTINC/
8DELETE.S DAM.DEO/
8DELETE.S DAM.OZDMS/
8DELETE.S DAM.SPLIT/
8DELETE.S DAM.CLADET/
8DELETE.S DAM.MAPRNT/
8DELETE.S DAM.IDFILE/
8DELETE.S DAM.LOCATE/
8DELETE.S DAM.OP3MOP/
8DELETE.S DAM.CONDIA/
8DELETE.S DAM.STATUS/

```

8MSO.N  SYS-GENCOM RUNSTREAM TO GENERATE SYS-COMPILE EXEC COMMAND RUNSTREAM
8MSO.N  -----
8MSO.N
8MSO.N
8MSO.N  HISTORY
8MSO.N  -----
8MSO.N
8MSO.N  E M SCHLOSSER      LEC      09/18/73      ORIGINAL CODE
8MSO.N  E M SCHLOSSER      LEC      07/31/79      ADD F OPTION TO ASH (QTR-WORD MODE
8MSO.N
8MSO.N  EXCEPTIONS
8MSO.N  -----
8MSO.N
8MSO.N  1. THIS ELEMENT IS DESIGNED FOR USE AT NASA/JSC ONLY TO GENERATE
8MSO.N     THE SYS-COMPILE ELEMENT ON PROGRAM TAPES DISTRIBUTED TO OTHER
8MSO.N     INSTALLATIONS.
8MSO.N
8MSO.N
8MSO.N  DAM PACKAGE SYSTEM GENERATION CONTROL STREAMS (SYS-GENCOM)
8SSO.KOP  .DAM./ELM.....PCF/1.DAM.
SKEL
*ELT.ID DAM.SYS-COMPILE
*INCREMENT NOUT TO IELM1      . **** PDP ALL PROCEDURE ELEMENTS ****
* IF IELM.NOUT.3.11 = *2
*   HDO DAM ASSEMBLER PROCEDURE IELM.NOUT.1.11/IELM.NOUT.2.11
*   POP.L DAM.IELM.NOUT.1.11/IELM.NOUT.2.11..IELM.NOUT.1.11/IELM.NOUT.2.11
*   END
* IF IELM.NOUT.3.11 = *3
*   HDO DAM COBOL PROCEDURE IELM.NOUT.1.11/IELM.NOUT.2.11
*   POP.LC DAM.IELM.NOUT.1.11/IELM.NOUT.2.11..IELM.NOUT.1.11/IELM.NOUT.2.11
*   END
* IF IELM.NOUT.3.11 = *4
*   HDO DAM FORTRAN PROCEDURE IELM.NOUT.1.11/IELM.NOUT.2.11
*   POP.LF DAM.IELM.NOUT.1.11/IELM.NOUT.2.11..IELM.NOUT.1.11/IELM.NOUT.2.11
*   END
*LOOP
*PACK.SR DAM.
*CLEAR NREL
*INCREMENT NOUT TO IELM1      . **** COMPILE SOURCE ELEMENTS ****
* IF IELM.NOUT.3.11 = *1      . TYPE 1 = SYMBOLIC
*   IF IELM.NOUT.3.21 > *1
*     HDO DAM IELM.NOUT.1.11/IELM.NOUT.2.11
*     IF IELM.NOUT.3.21 = *2      . SUBTYPE 2 = ASH
*       SET NREL TO NREL+1
*       ASH.FS DAM.IELM.NOUT.1.11/IELM.NOUT.2.11
*     END
*     IF IELM.NOUT.3.21 = *3      . SUBTYPE 3 = COB
*       SET NREL TO NREL+1
*       COB.LBKR DAM.IELM.NOUT.1.11/IELM.NOUT.2.11
*     END
*     IF IELM.NOUT.3.21 = *4      . SUBTYPE 4 = FOR
*       SET NREL TO NREL+1
*       FOR.S DAM.IELM.NOUT.1.11/IELM.NOUT.2.11
*     ADD DAM.SYS-FOROPT
*   END

```

DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-GENCON
002

```
•      IF *NREL > *20
•      CLEAR NREL
•      PACK DAM.
•      END
•      END
•      END
•LOOP
•END
•ELT.10 DAM.SYS-DELETE
•INCREMENT NOUT TO IELM) . **** DELETE SOURCE SYMBOLIC ELEMENTS ****
•  IF IELM.NOUT.3.1) = *1
•    IF IELM.NOUT.3.2) > *1
•      IF IELM.NOUT.3.2) < *6
•        DELETE.S DAM.IELM.NOUT.1.1)/IELM.NOUT.2.1)
•      END
•    END
•  END
•  END
•LOOP
•END
SEOF
SEOF
BCYCLE DAM.SYS-COMPILE.1
SED.U DAM.SYS-COMPILE
C / //A
EXIT
BCYCLE DAM.SYS-DELETE.1
SED.U DAM.SYS-DELETE
C / //A
EXIT
```

PREFACE TO APPENDIX L

THIS APPENDIX CONTAINS THE MAIN PROGRAMS (INCLUDING PSEUDO EXEC COMMANDS) IN THE DAN PACKAGE AND THEIR DEDICATED ROUTINES, GROUPED BY PROGRAM AS FOLLOWS:

MAIN PROGRAM (REAL VERSION)
HIERARCHY
MAIN PROGRAM (VIRTUAL VERSION)
COLLECTOR (LINKER) DIRECTIVES FOR REAL VERSION
COLLECTOR (LINKER) DIRECTIVES FOR VIRTUAL VERSION
DEDICATED COMMAND ROUTINES
DEDICATED UTILITY ROUTINES

THE NAMES OF MOST DEDICATED ROUTINES CONTAIN A THREE-LETTER PREFIX (IF COMMAND) OR SUFFIX (IF UTILITY) WHICH INDICATES THE PROGRAM THEY ARE DEDICATED TO:

PIC	PICTAB
CON	CONTROL
CLA	CLASSIFY
PRD	PRTOET
PRC	PRTCLAS
STA	STATUS
DIT	DITCOP

THE LAST THREE CHARACTERS OF DEDICATED COMMAND ROUTINES ARE COMPOSED OF THE FIRST THREE (OR TWO IF NOT PHASE 0) CHARACTERS OF THE COMMAND KEY WORD.

THE MORE COMPLEX DEDICATED COMMANDS WHICH REQUIRE LARGE AMOUNTS OF MEMORY FOR BUFFERS AND WORKSPACE ARE USUALLY BROKEN INTO SEVERAL ROUTINES TO PROCESS THE DIFFERENT PHASES. THE LAST CHARACTER OF A DEDICATED COMMAND ROUTINE INDICATES ITS PHASE NUMBER AS FOLLOWS:

LETTER	PHASE 0	(GET SPECIFICATIONS)
'1', '8'	PHASE 1, 2	(ADDITIONAL PRE-PROCESSING)
'3', '4', '5'	PHASE 3, 4, 5	(MAJOR PROCESSING -- FORTRAN I/O NOT ALLOWED)
'9'	PHASE 9	(POST-PROCESSING)

COMMAND ROUTINES IN PHASE 0 ARE GROUPED UNDER A CONVERSATIONAL MONITOR WHOSE NAME ENDS IN '000'. DEDICATED COMMAND ROUTINES IN OTHER PHASES ARE GROUPED BY PHASE UNDER THE '129' AND '349' MONITORS. NORMALLY THESE THREE MONITORS (AND THEIR COMMAND ROUTINES) OVERLAY EACH OTHER. SINCE DIRECT CALLS BETWEEN ROUTINES UNDER DIFFERENT MONITORS ARE NOT POSSIBLE, THE 'CALLING' ROUTINE CALLS NVIATO, NAMING THE 'CALLED' MONITOR AS THE 'VIA' ROUTINE AND THE 'CALLED' COMMAND PHASE ROUTINE AS THE 'TO' ROUTINE, AND THEN RETURNS TO ITS OWN MONITOR. ITS MONITOR WILL THEN RETURN TO VIATO, AND VIATO WILL THEN CALL THE PREVIOUSLY NAMED 'VIA' ROUTINE, PASSING IT THE NAME (ACTUALLY THE ADDRESS) OF THE PREVIOUSLY NAMED 'TO' ROUTINE, WHICH THE 'VIA' MONITOR WILL THEN CALL.

BEFORE RETURNING, ANY 'TO' ROUTINE WHICH IS NOT PHASE 0 MUST CALL NVIATO AND NAME THE NEXT MONITOR AND COMMAND PHASE ROUTINE. OTHERWISE IT WILL BE CALLED REPEATEDLY IN AN ENDLESS LOOP (THIS IS EASY TO DETECT WITH TRACE TURNED ON). THE LAST COMMAND PHASE ROUTINE FOR THE CURRENT COMMAND (UNLESS IT IS PHASE 0) MUST CALL NVIATO, NAMING THE '000' MONITOR AS THE

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

U
PREFACE-L
002

'VIA' ROUTINE AND NULSUB AS THE 'TO' ROUTINE. THE '000' MONITOR WILL THEN READ THE NAME OF THE NEXT COMMAND AND CALL THE APPROPRIATE PHASE 0 COMMAND ROUTINE.

DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

APPENDIX-L
 001

@PRT.SC DAM.PREFACE-L . (0009) SET TABS @ 30 & 34
 @PRT.S DAM.APPENDIX-L
 @MSG.N
 @PRT.S DAM.SETUP . SET UP TPF'S AND TTY. AND PRINT NEWS
 @PRT.SC DAM.SETUP-HIA . HIERARCHY
 @MSG.N
 @PRT.S DAM.SETUP/N . SET UP TPF'S AND TTY. AND DON'T PRINT NEWS
 @MSG.N
 @PRT.S DAM.ERSPRTCN . SUBMIT PRINT CONTROL SPECIFICATIONS
 @PRT.SC DAM.ERSPRTCN-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
 @PRT.SC DAM.ERSPRTCN-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
 @MSG.N
 @PRT.S DAM.DATA/CHECKOUT . PSEUDO EXEC COMMAND: INITIATE DATA/CHECKOUT MODE
 @PRT.SC DAM.DATA-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
 @PRT.SC DAM.DATA-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
 @PRT.SC DAM.DATA-CHECK . @DATA/CHECKOUT
 @MSG.N
 @PRT.S DAM.IDFILE . PSEUDO EXEC COMMAND: IDENTIFY TAPE OR DISK FILE
 @PRT.SC DAM.IDFILE-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
 @PRT.SC DAM.IDFILE-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
 @MSG.N
 @PRT.S DAM.LOCATE . PSEUDO EXEC COMMAND: POSITION MULTI-FILE TAPE
 @PRT.SC DAM.LOCATE-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
 @PRT.SC DAM.LOCATE-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
 @MSG.N
 @PRT.S DAM.USWAP . PSEUDO EXEC COMMAND: SWAP TAPE DRIVE UNITS
 @PRT.SC DAM.USWAP-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
 @PRT.SC DAM.USWAP-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
 @MSG.N
 @PRT.S DAM.ERTS-DUP . DUPLICATE ERTS MSS TAPE
 @MSG.N
 @PRT.S DAM.ERTSIDC . PRINT ID/ANNOTATION DATA FROM ERTS MSS TAPES
 @PRT.SC DAM.ERTSIDC-HIA . HIERARCHY
 @PRT.SC DAM.ERTSIDC/VIRTUAL . @MAP DAM.ERTSIDC TO TPF'S AND @XQT.1
 @PRT.SC DAM.ERTSIDC-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
 @PRT.SC DAM.ERTSIDC-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
 @MSG.N
 @PRT.S DAM.PICTAB . DISPLAY/TABULATE/FACTOR/PARTITION ERTS MSS DATA
 @PRT.SC DAM.PICTAB-HIA . HIERARCHY
 @PRT.SC DAM.PICTAB/VIRTUAL . @MAP DAM.PICTAB TO TPF'S AND @XQT.1
 @PRT.SC DAM.PICTAB-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
 @PRT.SC DAM.PICTAB-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
 @PRT.SC DAM.PIC000 . CALL PHASE 0 (COMMAND) ROUTINES FOR PICTAB
 @PRT.SC DAM.PIC129 . CALL PHASE 1/2/9 ROUTINES FOR PICTAB
 @PRT.SC DAM.PIC345 . CALL PHASE 3/4/5 ROUTINES FOR PICTAB
 @PRT.SC DAM.PIC678 . CALL PHASE 6/7/8 ROUTINES FOR PICTAB
 @PRT.SC DAM.PICDIS . DISPLAY MSS-DERIVED DATA (PHASE 0)
 @PRT.SC DAM.PICDI3 . DISPLAY RADIANCE (PHASE 3)
 @PRT.SC DAM.PICDI4 . DISPLAY GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)
 @PRT.SC DAM.PICDI5 . DISPLAY CLASS (PHASE 5)
 @PRT.SC DAM.PICDI9 . DISPLAY MSS-DERIVED DATA (PHASE 9)
 @PRT.SC DAM.DISHIS . HISTOGRAM DISPLAYED DATA (PHASE 0)
 @PRT.SC DAM.PICEXI . TERMINATION ROUTINE (PHASE 0)
 @PRT.SC DAM.PICFAC . FACTOR MSS CHANNELS (PHASE 0)
 @PRT.SC DAM.PICFA3 . FACTOR MSS CHANNELS (PHASE 3)
 @PRT.SC DAM.PICFA9 . FACTOR MSS CHANNELS (PHASE 9)

BPRT.SC DAM.PICLI3	. LIST MSS-DERIVED DATA (PHASE 0)
BPRT.SC DAM.PICLI3	. LIST RADIANCE (PHASE 3)
BPRT.SC DAM.PICLI4	. LIST GRADIENT (PHASE 4)
BPRT.SC DAM.PICLI5	. LIST CLASS (PHASE 5)
BPRT.SC DAM.PICLI9	. LIST MSS-DERIVED DATA (PHASE 9)
BPRT.SC DAM.PICPAR	. PARTITION FACTOR SPACE (PHASE 0)
BPRT.SC DAM.PICPA3	. PARTITION BY DENSITY (PHASE 3)
BPRT.SC DAM.PICPA4	. PARTITION BY GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)
BPRT.SC DAM.PICPA6	. PARTITION FACTOR SPACE (PHASE 6)
BPRT.SC DAM.PICPA9	. PARTITION FACTOR SPACE (PHASE 9)
BPRT.SC DAM.PICPIC	. PICTURE MSS-DERIVED DATA (PHASE 0)
BPRT.SC DAM.PICPI3	. PICTURE RADIANCE (PHASE 3)
BPRT.SC DAM.PICPI4	. PICTURE GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)
BPRT.SC DAM.PICPI5	. PICTURE CLASS (PHASE 5)
BPRT.SC DAM.PICPI9	. PICTURE MSS-DERIVED DATA (PHASE 9)
BPRT.SC DAM.PICPRO	. PROFILE MSS-DERIVED DATA (PHASE 0)
BPRT.SC DAM.PICPR3	. PROFILE MSS-DERIVED DATA (PHASE 3)
BPRT.SC DAM.PICPR9	. PROFILE MSS-DERIVED DATA (PHASE 9)
BPRT.SC DAM.PICROT	. ROTATE FACTOR STRUCTURE/COEFFICIENTS
BPRT.SC DAM.PICTOT	. TOTAL TABULATIONS
BPRT.SC DAM.PICTO3	. TOTAL TABULATIONS
BPRT.SC DAM.PICTO9	. TOTAL TABULATIONS
BPRT.SC DAM.PICXQT	. INITIALIZATION ROUTINE (PHASE 0)
BPRT.SC DAM.IDCPIC	. IDENTIFY CURRENT COMMAND SPECS (UTILITY)
BPRT.SC DAM.OPRPIC	. OPEN ALTERNATE PRINT FILES (UTILITY)
BMSG.N	
BPRT.S DAM.CONTROL	. ADJUST/DIAGRAM CONTROL NETWORK
BPRT.SC DAM.CONTROL-HIA	. HIERARCHY
BPRT.SC DAM.CONTROL/VIRTUAL	. BMAP DAM.CONTROL TO TPF'S AND BXQT.I
BPRT.SC DAM.CONTROL-MAP	. COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPRT.SC DAM.CONTROL-MAP/VIRTUAL	. COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
BPRT.SC DAM.CON000	. CALL PHASE 0 (COMMAND) ROUTINES FOR CONTROL
BPRT.SC DAM.CONADJ	. ADJUST NETWORK (PHASE 0)
BPRT.SC DAM.CONDIA	. DIAGRAM NETWORK (PHASE 0)
BPRT.SC DAM.CONEXI	. TERMINATION ROUTINE (PHASE 0)
BPRT.SC DAM.CONXQT	. INITIALIZATION ROUTINE (PHASE 0)
BPRT.SC DAM.DIAERR	. DIAGRAM ERRORS
BPRT.SC DAM.OLSTSQ	. LEAST SQUARES 81-LINEAR FIT
BMSG.N	
BPRT.S DAM.CLASSIFY	. CLASSIFY DATA ON ERTS MSS TAPE
BPRT.SC DAM.CLASSIFY-HIA	. HIERARCHY
BPRT.SC DAM.CLASSIFY/VIRTUAL	. BMAP DAM.CLASSIFY TO TPF'S AND BXQT.I
BPRT.SC DAM.CLASSIFY-MAP	. COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPRT.SC DAM.CLASSIFY-MAP/VIRTUAL	. COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
BPRT.SC DAM.CLA000	. CALL PHASE 0 (COMMAND) ROUTINES FOR CLASSIFY
BPRT.SC DAM.CLA129	. CALL PHASE 1/2/9 ROUTINES FOR CLASSIFY
BPRT.SC DAM.CLA345	. CALL PHASE 3/4/5 ROUTINES FOR CLASSIFY
BPRT.SC DAM.CLADET	. GENERATE DETECTION FILE (PHASE 0)
BPRT.SC DAM.CLADE3	. GENERATE RADIANCE DETECTION FILE (PHASE 3)
BPRT.SC DAM.CLADE4	. GENERATE DENSITY DETECTION FILE (PHASE 4)
BPRT.SC DAM.CLADE5	. GENERATE CLASS DETECTION FILE (PHASE 5)
BPRT.SC DAM.CLADE9	. GENERATE DETECTION FILE (PHASE 9)
BPRT.SC DAM.CLAEXT	. TERMINATION ROUTINE (PHASE 0)
BPRT.SC DAM.CLARAD	. GET/CHECK RADIANCE LIMITS (PHASE 0)
BPRT.SC DAM.CLAXQT	. INITIALIZATION ROUTINE (PHASE 0)
BPRT.SC DAM.CLS02N	. CLOSE OUTPUT DETECTION FILE (UTILITY)

DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

APPENDIX-L
 003

BPRT.SC DAM.OPNO2N	. OPEN OUTPUT DETECTION FILE (UTILITY)
BPRT.SC DAM.OPRCLA	. OPEN ALTERNATE PRINT FILE (UTILITY)
BPRT.SC DAM.SLMCLA	. PLOT SPECTRAL LIMITS (UTILITY)
BMSO.N	
BPRT.S DAM.PRTDET	. DISPLAY DATA FROM DETECTION FILE(S)
BPRT.SC DAM.PRTDET-HIA	. HIERARCHY
BPRT.SC DAM.PRTDET/VIRTUAL	. BMAP DAM.PRTDET TO TPF8 AND BXQT.1
BPRT.SC DAM.PRTDET-MAP	. COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPRT.SC DAM.PRTDET-MAP/VIRTUAL	. COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
BPRT.SC DAM.PRO000	. CALL PHASE 0 (COMMAND) ROUTINES FOR PRTDET
BPRT.SC DAM.PRO129	. CALL PHASE 1/2/9 ROUTINES FOR PRTDET
BPRT.SC DAM.PRO349	. CALL PHASE 3/4/5 ROUTINES FOR PRTDET
BPRT.SC DAM.PRO015	. DISPLAY DETECTION DATA (PHASE 0)
BPRT.SC DAM.PRO013	. DISPLAY DETECTION DATA (PHASE 3)
BPRT.SC DAM.PRO019	. DISPLAY DETECTION DATA (PHASE 9)
BPRT.SC DAM.PRDEXI	. TERMINATION ROUTINE
BPRT.SC DAM.PROL15	. LIST DETECTION DATA (PHASE 0)
BPRT.SC DAM.PROL13	. LIST DETECTION DATA (PHASE 3)
BPRT.SC DAM.PROL19	. LIST DETECTION DATA (PHASE 9)
BPRT.SC DAM.PROPIC	. PICTURE DETECTION DATA (PHASE 0)
BPRT.SC DAM.PROPI3	. PICTURE DETECTION DATA (PHASE 3)
BPRT.SC DAM.PROPI9	. PICTURE DETECTION DATA (PHASE 9)
BPRT.SC DAM.PRDXTQ	. INITIALIZATION ROUTINE
BPRT.SC DAM.IDCPRD	. IDENTIFY CURRENT COMMAND SPECS (UTILITY)
BPRT.SC DAM.OPRPRO	. OPEN ALTERNATE PRINT FILES (UTILITY)
BMSO.N	
BPRT.S DAM.PRTCLASS	. OUTPUT CLASSIFIED ERTS MAPS ON LINE PRINTER
BPRT.SC DAM.PRTCLASS-HIA	. HIERARCHY
BPRT.SC DAM.PRTCLASS/VIRTUAL	. BMAP DAM.PRTCLASS TO TPF8 AND BXQT.1
BPRT.SC DAM.PRTCLASS-MAP	. COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPRT.SC DAM.PRTCLASS-MAP/VIRTUAL	. COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
BPRT.SC DAM.PRC000	. CALL PHASE 0 (COMMAND) ROUTINES FOR PRTCLASS
BPRT.SC DAM.PRCEXI	. TERMINATION ROUTINE (PHASE 0)
BPRT.SC DAM.PRCMAP	. MAP RADIANCE/DENSITY/CLASS (PHASE 0)
BPRT.SC DAM.PRCXQT	. INITIALIZATION ROUTINE (PHASE 0)
BPRT.SC DAM.MAPRNT	. PRINT MAPS
BPRT.SC DAM.IITHOG	. PROVIDE UNIT HEADING (UTILITY)
BPRT.SC DAM.OPRPRC	. OPEN ALTERNATE PRINT FILES (UTILITY)
BPRT.SC DAM.RESPRC	. RESAMPLE DETECTION PIXELS (UTILITY)
BMSO.N	
BPRT.S DAM.PLTCLASS	. OUTPUT CLASSIFIED ERTS MAPS ON PEN PLOTTER
BPRT.SC DAM.PLTCLASS/VIRTUAL	. BMAP DAM.PLTCLASS TO TPF8 AND BXQT.1
BPRT.SC DAM.PLTCLASS-MAP	. COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPRT.SC DAM.PLTCLASS-MAP/VIRTUAL	. COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
BPRT.SC DAM.PLC000	. CALL PHASE 0 (COMMAND) ROUTINES FOR PLTCLASS
BPRT.SC DAM.PLC129	. CALL PHASE 1/2/9 ROUTINES FOR PLTCLASS
BPRT.SC DAM.PLCXQT	. TERMINATION ROUTINE (PHASE 0)
BPRT.SC DAM.PLCMAP	. MAP RADIANCE/DENSITY/CLASS (PHASE 0)
BPRT.SC DAM.PLCXQT	. INITIALIZATION ROUTINE (PHASE 0)
BPRT.SC DAM.ITICPL	. GENERATE & PLOT INTERIOR TICKS (UTILITY)
BPRT.SC DAM.MTICPL	. GENERATE & PLOT MARGINAL TICKS (UTILITY)
BPRT.SC DAM.NITHPL	. PLOT UNIT HEADING (UTILITY)
BPRT.SC DAM.OPLPLC	. OPEN ALTERNATE PLOT FILES (UTILITY)
BPRT.SC DAM.REGPLC	. REGISTER DETECTION PIXELS (UTILITY)
BMSO.N	
BPRT.S DAM.FLMCLASS	. OUTPUT CLASSIFIED ERTS MAPS ON FILM RECORDER

ORIGINAL PAGE IS
 OF POOR QUALITY

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

APPENDIX-L
004

0MS0.N
0PRT.S DAM.STATUS . DETERMINE STATUS OF DAM PACKAGE RUNS
0PRT.SC DAM.STATUS-HIA . HIERARCHY
0PRT.SC DAM.STATUS/VIRTUAL . 0MAP DAM.STATUS TO TPFS & 0XQT.1
0PRT.SC DAM.STATUS-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
0PRT.SC DAM.STATUS-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
0MS0.N
0PRT.S DAM.DITCOP . DISK TO TAPE COPY PROGRAM
0PRT.SC DAM.DITCOP/VIRTUAL . 0MAP DAM.DITCOP TO TPFS AND 0XQT.1
0PRT.SC DAM.DITCOP-MAP . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
0PRT.SC DAM.DITCOP-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
0PRT.SC DAM.DIT000 . CALL PHASE 0 (COMMAND) ROUTINES FOR DITCOP
0PRT.SC DAM.DITEX1 . TERMINATION ROUTINE
0PRT.SC DAM.DIT0UP . DUPLICATE DETECTION FILE ONTO TAPE FROM DISK
0PRT.SC DAM.DITVER . VERIFY DETECTION FILE ON TAPE
0PRT.SC DAM.DITXQT . INITIALIZATION ROUTINE

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

SETUP
001

@SETUPS: FREE TPFS. . (TOO SMALL) . :SETUPS
@SETUPS: ASO.T TPFS..F/O/TRK/256 . (BIGGER) . :SETUPS
@SETUPS: COPY.A DAM..TPFS. . :SETUPS
@SETUPS: XQT ERSPTCN . :SETUPS
D.00TY C.010
D.00TY W.132

LINE DELETE IS CTRL-X
BACKSPACE IS CTRL-H
TERMINAL IS 80 COLUMNS WIDE

@SETUPS: ADD.E DAM.NEN-DAM . :SETUPS

SETUP HIERARCHY

SETUP
:
:
:
ERSPTCN
:
:
:
NEH-DAM

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

SETUP/H
001

8SETUPS: FREE TPFS. . (TOO SMALL) . :SETUPS
8SETUPS: ASG.T TPFS..F/O/TRK/256 . (BIGGER) . :SETUPS
8SETUPS: COPY.A DAN..TPFS. . :SETUPS
8SETUPS: XQT ERSPRTCN . :SETUPS
D.88TTY C.010
D.88TTY H.132

LINE DELETE IS CTRL-X
BACKSPACE IS CTRL-H
TERMINAL IS 80 COLUMNS WIDE

8SETUPS: EOF . :SETUPS

PROGRAM ERSPRTCN 3 OUTPUT PRINT IMAGES / PERFORM PRINT CONTROL DIRECTIVES

HISTORY

E H SCHLOSSER	LEC	08/27/75	ORIGINAL CODE
E H SCHLOSSER	LEC	12/18/79	ELIMINATE THIRD-WORD J-DESIGNATORS

METHOD

THIS ASSEMBLER PROGRAM MAKES THE PRINT FACILITIES OF EXEC-8 ER PRINTS AND THE PRINT CONTROL FACILITIES OF EXEC-8 ER PRTCN'S DIRECTLY AVAILABLE FROM THE RUNSTREAM. THIS IS PARTICULARLY USEFUL FOR PRINTING IMAGES FROM AN 3ADD-ED ELEMENT AND/OR SUBMITTING DEMAND SYMBIONT (33) CONTROL STATEMENTS FROM AN 3ADD-ED ELEMENT.

PROCESSING OF INPUT TO ERSPRTCN IS DETERMINED BY COLUMN 1:

COLUMN 1	PROCESSING
D	SUBMIT TO ER PRTCN'S (DEMAND RUN ONLY)
A.I.L.H.M.R.S.W	SUBMIT TO ER PRTCN'S (DEMAND & BATCH RUNS)
(BLANK).+.0.1	PRINT VIA ER PRINTS (1ST CHR IS FORTRAN CARRIA E CONTROL
(OTHER)	GENERATES DIAGNOSTIC FROM SYMBIONT

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE EXEC-8 OPERATING SYSTEM USING 8-BIT FIELDATA CHARACTERS. IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES, DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER PRINTS	3 PRINT FIELDATA IMAGE
ER PRTCN'S	3 PERFORM FIELDATA PRINT CONTROL DIRECTIVE
ER EXITS	3 TERMINATE PROGRAM EXECUTION

EXCEPTIONS

1. THIS PROGRAM MUST NOT BE INITIATED AS A PROCESSOR!!

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

DAH PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

ERSPRTCN
 002

. (2 = REAL TIME
 . (3 = LOW EXEC
 . (4 = DEMAND
 . (5 = DEADLINE BATCH
 . (6 = BATCH

. LOCAL DECLARATIONS
 . -----

. AXRS
 \$ (00) . D-BANK
 PF FORM 12.6.18
 PKT PF 1.1.BUF
 LEN RES 1
 LOCBUF * BUF
 BUF RES 14
 DBATCH '000000' . (ONLY REMAINS '0' IF BATCH)

. PROCEDURE
 . -----

\$ (01) . I-BANK
 ERSPTCN TNE.U A4.4 . 4=DEMAND
 SZ DBATCH . DEMAND: REPLACE '0' WITH ZERO

READ LA A0.(EOF.BUF) .
 ER READS . READ CARD IMAGE
 SA.M2 A0.LEN . LENGTH OF IMAGE IN WORDS
 TNZ LEN . SKIP NI IF LEN NOT ZERO
 J ADV1 . NOT CONTROL IMAGE, SO PRINT!
 LA.S1 A1.BUF+0 . 1ST CHARACTER OF IMAGE
 TNE.U A1.'00000' . SKIP NI IF NOT SPACE
 J ADV1 . ' ' ==> ADVANCE 1 LINE BEFORE PRINTING
 TNE.U A1.'00000*' . SKIP NI IF NOT '*'
 J ADV0 . '*' ==> ADVANCE 0 LINES BEFORE PRINTING
 TNE.U A1.'000000' . SKIP NI IF NOT '0'
 J ADV2 . '0' ==> ADVANCE 2 LINES BEFORE PRINTING
 TNE.U A1.'000001' . SKIP NI IF NOT '1'
 J EJECT . '1' ==> EJECT BEFORE PRINTING
 TNE A1.DBATCH . SKIP NI IF NOT DBATCH
 J READ . DBATCH, SO IGNORE!

CONTROL L A0.LOCBUF
 LXI A0.LEN
 ER PRICNS . SUBMIT CONTROL IMAGE
 J READ

ADV0 LA.U A1.0 . ADVANCE 0 LINES BEFORE PRINTING
 J PRINT
 ADV1 LA.U A1.1 . ADVANCE 1 LINE BEFORE PRINTING
 J PRINT
 ADV2 LA.U A1.2 . ADVANCE 2 LINES BEFORE PRINTING
 J PRINT
 EJECT LA.XU A1.-0 . EJECT BEFORE PRINTING

DAN PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

ERSPRTCN
 003

PRINT	SA.S2	AI.PKT	. RIGHTMOST 6 BITS OF * LINES TO ADVANCE
	SSL	AI.B	. PREPARE TO STORE NEXT 6 BITS
	SA.S1	AI.PKT	. LEFTMOST 6 BITS OF * LINES TO ADVANCE
	SA.S3	AO.PKT	. LENGTH OF IMAGE IN WORDS
	LA.U	AI.'88888	. BLANK OUT ...
	SA.S1	AI.BUF+0 1ST CHARACTER
	LA	AO.PKT	
	ER	PRINTS	. PRINT
	J	READ	.
EOF	ER	EXITS	. TERMINATE PROGRAM
	END	ERSPRTCN	

**DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**ERSPTCN-NAP
001**

**IN DAM.ERSPTCN
LIB DAM.**

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

ERSPRTCN-MAP/VIRTUAL
001

IN DAM.ERSPRTCN
LIB DAM.

C PROCESSOR DATA/CHECKOUT DATA/CHECKOUT MODE: INTERCEPT/CHECK RUNSTREAM
C -----

C HISTORY
C -----

C E H SCHLOSSER LEC 12/09/77 REQUIREMENTS

C METHOD
C -----

C THIS PROCESSOR ALLOWS THE USER TO VERIFY THE SYNTAX OF BOTH EXEC COMMANDS
C AND PROGRAM COMMANDS FOR A COMPLETE DAM PACKAGE RUN WITHOUT ACTUALLY
C MOUNTING ANY TAPES OR PROCESSING ANY LANDSAT MSS DATA.

C THIS PROCESSOR INTERCEPTS AND READS ALL CARD IMAGE INPUT IN THE RUNSTREAM.
C IT CHECKS ALL IMAGES FOR VALID EXEC COMMANDS AND VALID SYNTAX, AND FLAGS
C ERRORS.

C WHEN THIS PROCESSOR ENCOUNTERS AN 'EXIT' EXEC COMMAND FOR ANY PROGRAM IN
C THE DAM PACKAGE, IT TERMINATES AND INITIATES THAT PROGRAM WITH THE
C REQUESTED OPTIONS PLUS A 'D' OPTION (DATA MODE) AND A 'C' OPTION (CHECKOUT
C MODE).

C THAT PROGRAM THEN READS CARD IMAGE INPUT FROM THE RUNSTREAM. IT CHECKS
C ALL IMAGES FOR VALID COMMANDS AND VALID SYNTAX, AND FLAGS ERRORS. IT DOES
C NOT ASSIGN WORK FILES OR GENERATE OUTPUT (OTHER THAN DIAGNOSTICS). WHEN
C THE PROGRAM TERMINATES, IT RE-INITIATES THE DATA/CHECKOUT PROCESSOR.

C DATA/CHECKOUT MODE CONTINUES, ALTERNATING BETWEEN THIS PROCESSOR AND
C PROGRAMS IN THE DAM PACKAGE UNTIL THE FOLLOWING CARD IMAGE IS READ:
C SEND

C RESTRICTIONS
C -----

- C 1. THE ABSOLUTE ELEMENT FOR THIS PROCESSOR MUST BE STORED IN THE DAM PROGRAM
C FILE IN REAL (NOT VIRTUAL) FORM.
- C 2. THE FOLLOWING CONTROL CARD MUST APPEAR IN THE RUN BEFORE ANY REFERENCE
C TO THIS PROCESSOR:
C ADD DAM.SETUP
- C 3. WHEN IN DATA/CHECKOUT MODE, SYNTAX ERRORS ON AN ADD EXEC COMMAND ABORT
C THE RUN.
- C 4. DATA/CHECKOUT MODE MAY NOT BE INITIATED OR TERMINATED FROM WITHIN AN
C EXECUTING PROGRAM.

C SYNTAX
C -----

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DATA/CHECKOUT
002

```
C      @DATA/CHECKOUT      . INITIATES DATA/CHECKOUT MODE
C      @END                . TERMINATES DATA/CHECKOUT MODE
C
C      EXTERNAL REFERENCES
C      -----
C
C      ERPRNT
C
C      GLOBAL DECLARATIONS
C      -----
C
C      NONE.
C
C      LOCAL DECLARATIONS
C      -----
C
C      (TO BE DETERMINED)
C
C      PROCEDURE
C      -----
C
C      CALL ERPRNT(1.5,'DATA/CHECKOUT NOT IMPLEMENTED')
C      STOP
C      END
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DATA-NAP
001

IN DAM.DATA/CHECKOUT
LIB DAM.

C-2

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DATA-MAP/VIRTUAL
001

IN DAM.DATA/CHECKOUT
LIB DAM.

**DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**DATA-CHECK
001**

SDATAS: DATA/CHECKOUT MODE . DATA/CHECKOUT MODE . DATA/CHECKOUT MODE

```
C      PROCESSOR IDFILE  3 PSEUDO EXEC COMMAND -- IDENTIFY TAPE/DISK FILE
C      -----
C
C      HISTORY
C      -----
C      E H SCHLOSSER      LEC      05/11/74      ORIGINAL CODE
C      E H SCHLOSSER      LEMSCO   07/22/80      DUMP NON-CHAR INFO IN OCTAL/BINARY
C
C      USAGE
C      -----
C      1. THE ABSOLUTE ELEMENT FOR THIS PROCESSOR MUST BE STORED IN THE DAM
C          PROGRAM FILE IN REAL (NOT VIRTUAL) FORM.
C
C      2. THE FOLLOWING CONTROL CARDS MUST APPEAR IN THE RUN BEFORE ANY
C          REFERENCE TO IDFILE:
C          3USE DAM.<NAME OF DAM PROGRAM FILE>
C          3PRT.T
C          3ASQ.A DAM.
C          3ADD DAM.SETUP
C
C      SYNTAX
C      -----
C      3IDFILE <INTERNAL FILE NAME>
C
C      METHOD
C      -----
C      GET FILE NAME FROM EXEC-8 INFOR BUFFER. GET FACILITY INFO FROM EXEC-8
C      FITEMS PACKET. PRINT IN CHARACTER. OCTAL. AND BINARY.
C
C      MACHINE-DEPENDENT CODE
C      -----
C      ASSUMES 6 CHARACTERS PER INTEGER. ASSUMES 6 BITS PER CHARACTER.
C      MANIPULATES UNIVAC EXEC-8 I/O AND FACILITY PACKETS.
C      USES UNIVAC FORTRAN V FIELD FUNCTION.
C
C      EXTERNAL REFERENCES
C      -----
C      SYSGET      3 GET RECORD FROM SYSIN RUNSTREAM
C      ERPRNT      3 WRITE CHARACTER BUFFER TO PRIMARY OUTPUT DEVICE
C      ERF1TH      3 RETRIEVE FACILITY ASSIGNMENT INFORMATION PACKET
C      CBINIT      3 INITIALIZE CHARACTER BUFFER
C      GETCHR      3 GET CHARACTER FROM CHARACTER STRING
C      GETICE      3 GET INTEGER-CHARACTER-EQUIVALENT FROM CHAR STRING
C      CB4CST      3 CHARACTER BUFFER FOR CHARACTER STRING
```

```
C      CB4IN      8 CHARACTER BUFFER FOR INTEGER
C
C
C EXCEPTIONS
C -----
C
C      1. THE FOLLOWING CONDITIONS GENERATE DIAGNOSTICS:
C          MODULE NOT INVOKED AS PROCESSOR IN TPFS.
C          FILE NAME NOT SPECIFIED
C          FILE NOT ASSIGNED
C
C
C GLOBAL DECLARATIONS
C -----
C
C      INCLUDE NULCST.LIST      8 DEFINE NULL CHARACTER STRING
C      INCLUDE ICBUF1.LIST     8 DECLARE CHARACTER BUFFER # 1
C      INCLUDE ICBUF2.LIST     8 DECLARE CHARACTER BUFFER # 2
C      INCLUDE ICBUF3.LIST     8 DECLARE CHARACTER BUFFER # 3
C      INCLUDE ASHDEF.LIST     8 DEFINE ASSEMBLER PARTIAL WORD MNEMONICS
C
C
C LOCAL DECLARATIONS
C -----
C
C      INTEGER INFOR(10)      8 UNIVAC EXEC-8 INFORS BUFFER
C      INTEGER IFPKT(13)      8 UNIVAC EXEC-8 FACILITY INFO PACKET
C      INTEGER INSTAT          8 INPUT STATUS FROM SYSOET
C      INTEGER LENGTH          8 INPUT LENGTH IN CHARS FROM SYSOET
C      INTEGER NW              8 WORD NUMBER, STARTING FROM 0 AT LEFT
C      INTEGER NC              8 CHARACTER NUMBER, STARTING FROM 1 AT LEFT
C      INTEGER NB              8 BIT NUMBER, STARTING FROM 0 AT LEFT
C      INTEGER KHAR            8 CHARACTER
C      INTEGER NICE            8 INTEGER-CHARACTER-EQUIVALENT
C      INTEGER IBIT            8 BIT
C
C
C PROCEDURE
C -----
C
C      GET UNIVAC EXEC-8 INFORS BUFFER CONTAINING EXEC COMMAND SPEC(S)
C
C      CALL SYSOET(INSTAT,INFOR,LENGTH)
C
C
C CHECK IF INVOKED AS A PROCESSOR IN TPFS.
C
C      IF((INSTAT.EQ.' ') .AND. (INFOR(3).EQ.'IDFILE')) GO TO 150
C          CALL ERPRNT(1,3,'PROGRAM NOT FOUND')
C          GO TO 900
C
C      150 CONTINUE
C
C
C CHECK IF FILE NAME WAS SPECIFIED
C
```

DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

IOFILE
 003

```

    IF(INFOR(5).NE.0) GO TO 200
      CALL ERPRNT(1.2,'NO FILE')
      GO TO 900
  200 CONTINUE
  C
  C
  C EXTRACT FILE NAME FROM INFORS BUFFER & PUT INTO PACKET
  C
    IFPKT(1)=INFOR(5)
    IFPKT(2)= ' '
    IF(ASHMSB(INFOR(4)).NE.1) IFPKT(2)=INFOR(6)
  C
  C
  C GET/CHECK EXEC-8 FACILITY INFO IN PACKET
  C
    CALL ERFITH(IFPKT)
    IF(IFPKT(7).NE.0) GO TO 300
      CALL ERPRNT(1.2,'NOT ASSIGNED')
      GO TO 900
  300 CONTINUE
  C
  C
  C PRINT FILE NAME & QUALIFIER FROM FITEMS PACKET WORDS 0 THRU 5
  C
    CALL ERPRNT(1.2,IFPKT(1))
    CALL ERPRNT(1.2,IFPKT(3))
    CALL ERPRNT(1.2,IFPKT(5))
  C
  C
  C DUMP PACKET WORDS 6 THRU 10 IN CHAR/OCTAL/BINARY
  C
    DO 460 NW=6,10
      CALL CBINIT(ICBUF1)
      CALL CB4CST(ICBUF1, 'FITEMS+')
      CALL CB4IN (ICBUF1, NW,2,'0')
      CALL CBINIT(ICBUF2)
      CALL CB4CST(ICBUF2, ICBUF1)
      CALL CBINIT(ICBUF3)
      CALL CB4CST(ICBUF3, ICBUF1)
      CALL CB4CST(ICBUF1, ' ',(1),(2))
      CALL CB4CST(ICBUF2, ' ',(1),(2))
      CALL CB4CST(ICBUF3, ' ',(1),(2))
    DO 430 NC=1,6
  C
      CALL GETCHR(KHAR, IFPKT(NW+1),(NC))
      CALL CB4CST(ICBUF1, KHAR,(1),(1))
      CALL CB4CST(ICBUF1, ' ',(1),(6))
  C
      CALL GETICE(NICE, IFPKT(NW+1),(NC))
      CALL CB4IN(ICBUF2, NICE/8,1)
      CALL CB4IN(ICBUF2, NICE-(NICE/8)*8,1)
      CALL CB4CST(ICBUF2, ' ',(1),(5))
  C
    DO 410 NB=0,5
      IBIT=FLD(NB,1,KHAR)
      CALL CB4IN(ICBUF3, IBIT,1)

```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IOFILE
004

```
410          CONTINUE
          CALL C04CST(1CBUF3,  * *,(11),(11))
430          CONTINUE
          CALL ERPRNT(2,10,1CBUF1)
          CALL ERPRNT(1,10,1CBUF2)
          CALL ERPRNT(1,10,1CBUF3)
460 CONTINUE
C
C
C PRINT TAPE REEL NUMBERS, IF PRESENT, FROM WORDS 12 & 13 OF PACKET
C
          IF(1FPKT(12).NE.NULCST) CALL ERPRNT(2,1,1FPKT(12))
          IF(1FPKT(13).NE.NULCST) CALL ERPRNT(1,1,1FPKT(13))
C
C
C TERMINATE
C
900 STOP
      END
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IDFILE-NAP
001

IN DAN.IDFILE
LIB DAN.

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IDFILE-MAP/VIRTUAL
001

IN DAN.IDFILE
LIB DAN.

C PROCESSOR LOCATE A PSEUDO EXEC COMMAND: LOCATE FILE ON MULTIFILE BIP TP
C -----

C HISTORY
C -----

C	E M SCHLOSSER	LEC	12/22/78	ORIGINAL CODE
C	E M SCHLOSSER	LEC	04/20/79	CHECK '-' IN SCENE NUMBER
C	E M SCHLOSSER	LEC	12/17/79	SUPPORT 88 OR 85T BUFFER FORMAT
C	E M SCHLOSSER	LEMSCO	07/22/80	REPLACE KHAR WITH GETCHR
C	E M SCHLOSSER	LEMSCO	09/27/80	CHK BUF FMT FROM FLINFO

C METHOD
C -----

C THIS PROCESSOR PROVIDES A PSEUDO EXEC COMMAND TO LOCATE LANDSAT STRIPS ON
C BIP ('X') FORMAT MULTI-FILE TAPE. IT IS A TEMPORARY COMPONENT OF THE DAM
C PACKAGE, DESIGNED FOR USE ONLY WITH THE INTERIM OGDARD MULTI-FILE BIP
C FORMAT CONTAINING 2 OR 4 DATA FILES AND 1 SIAT FILE ON A SINGLE REEL.

C RESTRICTIONS
C -----

- C 1. THE ABSOLUTE ELEMENT FOR THIS PROCESSOR MUST BE STORED IN THE DAM PROGRAM
C FILE IN REAL (NOT VIRTUAL) FORM.
- C 2. THE FOLLOWING CONTROL CARDS MUST APPEAR IN THE RUN BEFORE ANY REFERENCE
C TO LOCATE:
C SADD DAM.SETUP
C SASO.<OPTIONS> 3..U9.<REEL NUMBER>

C SYNTAX
C -----

C BLOCATE(.E) 3..<LANDSAT STRIP NUMBER>

C MACHINE-DEPENDENT CODE
C -----

C VERRY!!

C EXTERNAL REFERENCES
C -----

C	ERPCY	A GET PART OF EXEC-B PROGRAM CONTROL TABLE
C	EREXIT	A TERMINATE PROGRAM
C	ERFACL	A RETRIEVE FACILITIES ASSIGNMENT INFORMATION
C	ERPRINT	A PRINT IMAGE ON TTY OR LINE PRINTER (FIELDATA)
C	ERERR	A ERRS TERMINATE
C	ERREAD	A READ IMAGE FROM TTY OR CARD READER (FIELDATA)

C FLINFO & GET FILE INFORMATION
 C BST400 & INTERNAL BYTE STRING FOR EXTERNAL 8-BIT BYTE STRING
 C CST400 & CHARACTER STRING FOR EBCDIC BYTE STRING
 C GETBYT & GET BYTE FROM BYTE STRING
 C GETCHR & GET CHARACTER FROM CHARACTER STRING
 C ERTWAT & TIMED WAIT FOR UP TO 30 SECONDS
 C ERIO & INITIATE I/O
 C ERWAIT & WAIT FOR COMPLETION OF I/O
 C INTEGER ICE & INTEGER-CHARACTER-EQUIVALENT (FROM CST)

C
 C GLOBAL DECLARATIONS
 C -----

C
 C INTEGER JPCT(25) & EXEC-8 PROGRAM CONTROL TABLE (1ST 25 WDS)
 C INCLUDE XGDEF.LIST & DEFINE STATUS OF XGQT OPTIONS
 C INCLUDE ASMDEF.LIST & DEFINE UNIVAC ASSEMBLER PARTIAL WDS IN FORTRAN V
 C INCLUDE KOMIO.LIST & FORTRAN MANIPULATION OF ASSEMBLER I/O PACKETS
 C INCLUDE FIDEF.LIST & MNEMONICS FOR LOCATIONS IN FIDEF-FORMAT BUFFER

C
 C LOCAL DECLARATIONS
 C -----

C
 C INTEGER INAGE(14) & EXEC-8 PROCESSOR INFOR BUFFER
 C INTEGER IDFIL3(10) & FILE INFORMATION IN FIDEF FORMAT FOR FILE 3
 C INTEGER NPKT3(8)/'S'.0'/' & EXEC-8 TAPE I/O PACKET FOR FILE '3'
 C INTEGER IOBUF(40) & IO RECORD BUFFER
 C DATA IOBUF(1)/0/ & ELIMINATE SPURIOUS COMPILER DIAGNOSTICS
 C INTEGER IANBUF(80) & ANNOTATION RECORD BUFFER
 C DATA IANBUF(1)/0/ & ELIMINATE SPURIOUS COMPILER DIAGNOSTICS
 C INTEGER R /'BK'/ & READ
 C INTEGER RB /'BL'/ & READ BACKWARD
 C INTEGER REW/'R'/ & REWIND

C
 C PROCEDURE
 C -----

C
 C RETRIEVE PROGRAM CONTROL TABLE & CONTROL CARD SPECIFICATIONS

CALL .RPCT(25,JPCT)
 CALL INFOR(8890)

C
 C CHECK FOR 'E' OPTION

IF(
 & (XGTOPT('ERROR').EQ.1).AND. & E OPTION SPECIFIED ...
 & (ASMT3(JPCT(17)).NE.2)) & PREVIOUS EXECUTION DID'NT ERROR TERMINATE
 & CALL EREXIT & ... THEN IGNORE THIS 'EXEC' COMMAND

C
 C CHECK IF TAPE IS ASSIGNED TO 3.
 C

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

LOCATE
003

```
CALL FLINFO(1DFIL3, '3', '00')
IF(1DFIL3(FIDEQT).EQ.'NUL') GO TO 010      & NOT ASSIGNED
IF(1DFIL3(FIDEQT).NE.'TAPE') GO TO 020      & NOT TAPE

C
C READ NEXT TAPE BLOCK AND CHECK IF IT IS 10 RECORD
C
CALL IDENT3(0300)
IF(1NCCT.LE.NSTRIP) GO TO 000

C
C REWIND TAPE, THEN READ & CHECK FIRST 10 RECORD
C
000 CALL 103(12W,0320)
020 CALL IDENT3(0340)
GO TO 000
040 CALL 103(12W,0360)
060 CALL IDENT3(0000)

C
C CHECK IF 10 RECORD IS FOR SPECIFIED STRIP
C
000 IF(1NCCT.EQ.NSTRIP) GO TO 000
IF(1NCCT.GT.NSTRIP) GO TO 030

C
C FIND NEXT END-OF-FILE MARK
C
CALL 103(1R,0000)
000 CALL 103(1R,0700)
GO TO 000

C
C READ BLOCK AFTER EOF & CHECK IF IT IS 10 RECORD
C
000 CALL IDENT3(0030)
GO TO 000

C
C FLAG TAPE ERROR(S)
C
000 CALL ERPRNT(1.2, 'TAPE ERROR')
GO TO 090
010 CALL ERPRNT(1.3, 'FILE NOT ASSIGNED')
GO TO 090
020 CALL ERPRNT(1.3, 'FILE NOT TAPE')
GO TO 090
030 CALL ERPRNT(1.3, 'STRIP NOT ON TAPE')
090 IF(1ASHS2(JPCT(25)).EQ.4) GO TO 090      & ONLY IN DEMAND
IF(1XOTOPT('CONTINUE').EQ.1) GO TO 090
CALL ERERR      & ERRS TERMINATION!

C
C POSITION TAPE BEFORE 10 RECORD
C
000 CALL 103(1R,0000)
```

DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

LOCATE
 884

CALL I03'R.8000)
 IF(IASH21JPC1(25).EQ.4) CALL ERPRNT(1.1,'READY') & ONLY IN DEMAND
 999 CALL EXRIT

C
 C
 C
 C
 C
 C
 C

SUBROUTINE INFOR(S)

C

CALL ERREAD(8890,IMAGE,NA0)
 IF(IMAGE(3).NE.'LOCATE') GO TO 899
 IF(IQTOPT('CONTIN').EQ.1) & C OPTION SPECIFIED ..
 & NPKT3(1)=IMAGE(5) & .. SO ANY FILENAME OK
 IF(IMAGE(5).NE.NPKT3(1)) GO TO 899
 IF(FLD(8,30,IMAGE(7)).NE.'8 ') GO TO 870
 NSTRIP=FLD(10,8,IMAGE(7))-'88888'
 IF(INSTRIP.LT.1).OR.(INSTRIP.GT.4)) GO TO 870
 RETURN

C
 C
 C
 C

FLAO SYNTAX ERRORS

859 CALL ERPRNT(1.2,'INVALID FILE')
 GO TO 899
 870 CALL ERPRNT(1.4,'INVALID STRIP NUMBER')
 GO TO 899
 890 CALL ERPRNT(1.7,'PROGRAM NOT FOUND')
 899 RETURN

C
 C
 C
 C
 C
 C

SUBROUTINE IDENT3(S)

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

READ (HOPEFULLY) ID RECORD
 IDBUF(10)=0
 IOSIZE(INPKT3)=40
 IOADDR(INPKT3)=LOC(IDBUF)
 CALL I03'R.8000)
 IF(IONWDS(INPKT3).GT.14) GO TO 800 & TOO LONG TO BE AN ID RECORD
 IF(IDFIL3(IDBFH).NE.'88') GO TO 300
 DO 200 NBY=1,30
 CALL GETBYT(1BYT, IDBUF.(NBY))
 IF(1BYT.GT.255) IDFIL3(IDBFH)='88'
 200 CONTINUE
 IF(IDFIL3(IDBFH).NE.'88') CALL ERPRNT
 & 1.6.'**88' BUF FMT FROM FLINFO WRONG**'
 300 IF(IDFIL3(IDBFH).EQ.'88') CALL 88*88(IDBUF, IDBUF.80)
 CALL CST*88(IDBUF, IDBUF.80)

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

LOCATE
005

C
C
C
C

READ ANNOTATION RECORD

```
LOADDR(NPKT3)=LOC(IANB.  
CALL 103(R,$800)  
IF(1DFIL3(F10BFH).EQ.'88') CALL 8ST4B8(IANBUF. IANBUF.300)  
CALL CST4EB(IANBUF. IANBUF.300)  
IF(XQTOPT('LIST').EQ.0) GO TO 600      8 L OPTION MEANS LIST TAPE HEADERS  
CALL ERPRNT(1.7.IANBUF(1))  
CALL ERPRNT(1.9.IANBUF(1))  
CALL ERPRNT(1.12.IANBUF(10))
```

C
C
C
C

GET/CHECK STRIP NUMBER

```
800 CALL GETCHR(KHTEMP. IANBUF.5)  
IF(KHTEMP.NE.'-') CALL GETCHR(KHTEMP. IANBUF.6)  
IF(KHTEMP.NE.'-') GO TO 800  
CALL GETICE(NICE. IANBUF.14)  
NCCT=NICE-ICE('0')  
IF((NCCT.LT.1).OR.(NCCT.GT.4)) GO TO 800  
CALL GETCHR(KHTEMP. IANBUF.15)  
IF(KHTEMP.NE.' ') GO TO 800  
CALL GETICE(NICE. IANBUF.16)  
NCCTOT=NICE-ICE('0')  
IF(NCCTOT.NE.4) GO TO 800  
RETURN      8 NORMAL RETURN  
800 RETURN 1      8 ERROR RETURN
```

C
C
C
C
C
C

SUBROUTINE 103(NXTFUN.5)

C

```
IF(NXTFUN.NE.LSTFUN) CALL ERWHAT(2000)      8 REVERSE TAPES GENTLY!!  
IOFUNC(NPKT3)=NXTFUN  
LSTFUN=NXTFUN  
CALL ERIO(NPKT3)  
IF(1OFUNC(NPKT3).GT.0) CALL ERWHAT(10)      8 FIX FOR EXEC PROBLEM  
CALL ERWAIT(NPKT3)  
IF(1OCODE(NPKT3).EQ.'EOF') RETURN 2      8 EOF ENCOUNTERED  
RETURN
```

C
C

END

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

LOCATE-MAP
001

IN DAH.LOCATE
LIB DAH.

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

LOCATE-MAP/VIRTUAL
001

IN DAN.LOCATE
LIB DAN.

```
C   PROGRAM USMAP 8 SWAP TAPE DRIVE UNITS
C   -----
C
C   HISTORY
C   -----
C       E H SCHLOSSER   LEC   12/09/77   REQUIREMENTS
C
C   METHOD
C   -----
C
C       THIS PROCESSOR ALLOWS THE USER TO SWAP TAPE DRIVE UNITS. USMAP
C       GENERATES ALL THE NECESSARY EXEC COMMANDS. WRITES THEM TO TEMPORARY
C       FILE 20. AND THEN 8ADD-S FILE 20 TO THE RUNSTREAM.
C
C   RESTRICTIONS
C   -----
C
C   1. THE ABSOLUTE ELEMENT FOR THIS PROCESSOR MUST BE STORED IN THE DAM PROGRAM
C       FILE IN REAL (NOT VIRTUAL) FORM.
C
C   2. THE FOLLOWING CONTROL CARD MUST APPEAR IN THE RUN BEFORE ANY REFERENCE
C       TO THIS PROCESSOR:
C           8ADD DAM.SETUP
C
C   3. A TAPE REEL MAY NOT BE SWAPPED MORE THAN ONCE IN ANY RUN.
C
C   4. IF TWO TAPE FILES ARE SPECIFIED. THEY MUST HAVE THE SAME DENSITY
C       AND NUMBER OF TRACKS.
C
C   SYNTAX
C   -----
C
C       8USWAP(.E) <TAPE FILE NAME>[.<TAPE FILE NAME>]
C
C   EXTERNAL REFERENCES
C   -----
C
C       ERPRINT
C
C   GLOBAL DECLARATIONS
C   -----
C
C       NONE.
C
C   LOCAL DECLARATIONS
C   -----
C
C       (TO BE DETERMINED)
```

BAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

USMAP
002

C
C
C PROCEDURE
C -----
C
C

CALL ERPRINT(1.4, 'USMAP NOT IMPLEMENTED')
STOP
END

ORIGINAL PAGE IS
OF POOR QUALITY

**DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**USMAP-MAP
001**

**IN DAM.USMAP
LIB DAM.**

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

USMAP-MAP/VIRTUAL
001

IN DAH.USMAP
LIB DAH.

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

ERTS-DUP
001

```

8HDD DAM ERTS-DUP(7703)
8JUMP BEGIN
8USAGE: . THIS EXEC 8 COMMAND STREAM COPIES ERTS TAPE INN TO OUT, IGNORING
8USAGE: . FRAME COUNT ERRORS. IF UNRECOVERABLE PARITY ERRORS OCCUR IT
8USAGE: . REWINDS BOTH TAPES AND TRIES AGAIN. IF AND WHEN AN APPARENTLY
8USAGE: . SUCCESSFUL COPY IS MADE, THE COPY IS IDENTIFIED AND VERIFIED, AND
8USAGE: . THE RUN TERMINATED. (E M SCHLOSSER)
8USAGE: .
8USAGE: . SEE DAM.EXP-ERTS-DUP FOR INSTRUCTIONS
8USAGE: . SEE DAM.RUN-ERTS-DUP FOR SAMPLE RUNSTREAM
8USAGE: .
8BEGIN:
8SETC.1 . DON'T ABORT RUN ON ERRS TERMINATION
8USE 3.INN.
8REWIND 3.
8XQT ERTSIDC
8REWIND INN.
8REWIND OUT.
8LOG DAM ERTS-DUP(7703) -- FIRST TRY
8COPY.MN INN..OUT..9 . MUST HAVE 2342 BLOCKS PER DATA FILE **** FIRST TRY ****
8TEST T0/1/52
8JUMP OK
8REWIND 3.
8XQT.E ERTSIDC
8REWIND INN.
8REWIND OUT.
8LOG DAM ERTS-DUP(7703) -- SECOND TRY
8COPY.MN INN..OUT..9 . MUST HAVE 2342 BLOCKS PER DATA FILE **** SECOND TRY ****
8TEST T0/1/52
8JUMP OK
8SHAP:
8TEST TE/0/T2
8JUMP END . (ONLY SHAP ONCE)
8USE OLDINN.INN.
8USE OLDOUT.OUT.
8USE INN.NEWINN.
8USE OUT.NEWOUT.
8JUMP END
8OK:
8FREE INN.
8CLOSE OUT.
8USE 3.OUT.
8XQT PICTAB
EXIT
8FREE OUT.
8FIN
8END:
8SETC.A 1/T2 . RESET 8 FLAG
```

C PROGRAM ERTSIDC & PRINT ID/HEADER AND ANNOTATION RECORD INFO
C -----
C

C HISTORY
C -----
C

C E H SCHLOSSER LEC 05/29/73 ALGORITHM CODING
C J C CRISP LEC 10/24/79 REVISE TO USE CHAR BUFFER ROUTINES
C

C METHOD
C -----
C

C CALL OPENS TO READ ID/HEADER AND ANNOTATION RECORDS. CALL ROUTINES
C TO PRINT INFORMATION EXTRACTED AND COMPUTED. BUILD BUFFER TO
C PUT ERTS ID ON TAIL SHEET.
C

C ERTS CONVENTIONS FOR ATTITUDE AND HEADING:
C

C POSITIVE PITCH IS NOSE DOWN
C POSITIVE PITCH IS CLOCKWISE VIEWED FROM BEHIND
C POSITIVE YAW IS COUNTERCLOCKWISE VIEWED FROM ABOVE
C POSITIVE HEADING IS CLOCKWISE VIEWED FROM ABOVE
C

C MACHINE-DEPENDENT CODE
C -----
C

C UTILIZES UNIVAC EXEC & ER CSFS AND ER PRINTS
C

C EXTERNAL REFERENCES
C -----
C

C PSTART & INITIALIZE PROGRAM
C OPENS & OPEN INPUT MSS/RBV FILE ASSIGNED TO UNIT 3
C IOLUS & PRINT SHORT ID FOR LOGICAL UNIT 3
C IDERTS & PRINT COMPLETE ERTS SCENE ID
C CBINIT & INITIALIZE CHARACTER BUFFER
C CB4CST & CHARACTER BUFFER FOR CHARACTER STRING
C CB4IN & CHARACTER BUFFER FOR INTEGER
C ERCSF & SUBMIT EXEC COMMANDS
C PSTOP & PROGRAM TERMINATION
C

C EXCEPTIONS
C -----
C

C NONE
C

C GLOBAL DECLARATIONS
C -----
C

C INCLUDE KONXOT.LIST & COMMON PROGRAM SWITCHES.COUNTERS
C

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

ERTS10C
002

```
        INCLUDE KOMNER.LIST      @ COMMON ERTS SCENE PARAMETERS
        INCLUDE ICBUF1.LIST      @ DECLARE CHARACTER BUFFER * 1

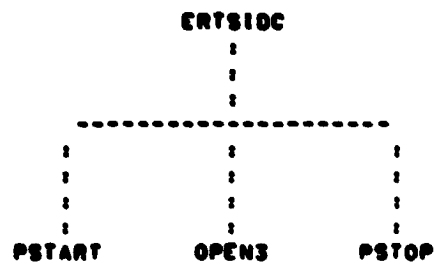
C
C LOCAL DECLARATIONS
C -----
C
C      NONE
C
C PROCEDURE
C -----
C
C INITIALIZE PROGRAM AND OPEN INPUT MSS FILE ON UNIT 3
C
C      CALL PSTART( 'DAM ERTS10C (0009)')
C      CALL OPEN 3
C      IF (MIDATAC.NE.0) GO TO 900

C
C IDENTIFY UNIT 3 HARDWARE/SENSOR/SCENE/DATE/ETC
C
C      CALL ERPRNT (1.1.' ')
C      CALL IDLUS (6)
C      CALL IDERTS (6)

C
C PUT SCENE ID INTO UNIVAC EXEC-8 SYSTEM LOG SO IT GETS PRINTED ON TAIL SHEET
C
C      CALL CBINIT (ICBUF1)
C      CALL CB4CST (ICBUF1, 'BLOG ',(1),(5))
C      CALL CB4CST (ICBUF1, NERSAT,(1),(12))
C      CALL CB4CST (ICBUF1, ' SCENE ',(1),(7))
C      CALL CB4IN (ICBUF1, NERTS(1),1)
C      CALL CB4IN (ICBUF1, NERTS(2),4,'0')
C      CALL CB4CST (ICBUF1, '- ')
C      CALL CB4IN (ICBUF1, NERTS(3),5,'0')
C      CALL CB4CST (ICBUF1, ' CCT ',(1),(7))
C      CALL CB4IN (ICBUF1, NCCT,1)
C      CALL CB4CST (ICBUF1, ' OF ',(1),(4))
C      CALL CB4IN (ICBUF1, NCCTOT,1)
C      CALL CB4CST (ICBUF1, ' . ',(1),(3))
C      CALL ERCSF (NAD,ICBUF1)

C
C TERMINATE
C
C 900 CALL PSTOP ('0***PLEASE BFREE 3. OR BREWIND 3. OR BLOCATE 3.')
C      END
```

ERTSIDC HIERARCHY



PROGRAM ERTSIDC/VIRTUAL

HISTORY

E H SCHLOSSER LEC 00/02/74 ORIGINAL CODE
E H SCHLOSSER LEC 11/06/79 SHAP.FZ(N); NO 'N' IN DEMAND)

METHOD

CONSTRUCT SHAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT SXQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE SHAP & SXQT COMMANDS TO TEMPORARY FILE 20.
ADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 8-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS 8 FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IOWS 8 INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS 8 TERMINATE PROGRAM EXECUTION
DAM.ERTSIDC-MAP 8 SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAOPT 8 STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS 8ASG-0 & 8PREP-0.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

 1 2 = REAL TIME
 1 3 = LOW EXEC
 1 4 = DEMAND
 1 5 = DEADLINE BATCH
 1 6 = BATCH

(SXQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER A5 IN

(MASTER BIT NOTATION.

LOCAL DECLARATIONS

```

      AXRS
S(00) . D-BANK
SSSH      FORM      0.0.0.10
          11111222223333334444455555666666777777888888999999
LABSDF    SSSH      000.1.'F'.0      . LABEL. 1 MD. FORTRAN. FIELDATA
LABING    "SOFF"
MAPSDF    SSSH      000.0.0.0      . DATA. 0 MDS. . FIELDATA
MAPING    'BXQTS: MAP.FZN DAN.ERTSIDC-MAP.ERTSIDC      . :BXQTS'
ADDSDF    SSSH      000.0.0.0
ADDING    'BXQTS: ADD      DAN.SYS-MAPOPT      . :BXQTS'
XQTSDF    SSSH      000.0.0.0
XQTING    'BXQTS: XQT.1      ERTSIDC      . :BXQTS'
EOFSDP    -      0      . END-OF-FILE STOP WORD
PF        FORM      12.0.10
CSFASO    'BASO.T 20. . .
CSFADD    'BADD 20. . .
SAVREG    RES      1
IOPKT     1000     '20'.MS 33.LABSDF.'0' 0
  
```

PROCEDURE

```

S(01) . I-BANK
ERTSIDC  LA,U      A0. .      . A0 10 .
          TNE,U    A4.4      . SKIP NEXT INST IF A4<>4 (NOT DEMAND)
          SA,52    A0.MAPING+2 . DEMAND! BLANK OUT N OPTION
          LA      A0.(CSFASO) . ADDRESS OF BASO IMAGE
          ER      CSFS      . DO IT
          SA      A0.SAVREG . STORE 6
          PPRINT  (PF 2.1.SAVREG) . PRINT BASO STATUS

          GETOPT      . LOAD OPT LTRS INTO A2.A3.A4

PUTOPT   DS      A2.XQTING+2 . STORE OPTION LETTERS INTO BXQT IMAGE
          SA      A4.XQTING+4 . (3 WORDS -- MAX 10 OPT LETTERS)

WRITE    LA      A0.(IOPKT) . ADDRESS OF I/O PACKET
          ER      10MS      . WRITE SDF IMAGES TO 20.

ADD      LA      A0.(CSFADD) . ADDRESS OF BADD IMAGE
          ER      CSFS      . DO IT
          ER      EXITS

END      ERTSIDC
  
```


**DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**ERTSIOC-NAP
001**

**IN DAN.ERTSIOC/..NTABS/DAN..SYS-BLOCK
LIB DAN.**

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

ERTSIOC-NAP/VIRTUAL
001

IN DAN.ERTSIOC/VIRTUAL

C PROGRAM PICTAB & PICTURE/TABULATE/DISPLAY/LIST/FACTOR NSS DATA
C -----
C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 07/02/73 ORIGINAL CODE
C J C CRISP LEHSCO 05/28/80 UPGRADE DOCUMENTATION & ADD KOMKS
C
C
C METHOD
C -----
C
C THIS PROGRAM PICTURES, DISPLAYS, LISTS, TABULATES,
C CORRELATES, FACTORS, MODELS, AND/OR PARTITIONS RAW OR
C TRANSFORMED DATA FROM LANDSAT BIP('X') OR BIL('AM' OR 'PH')
C COMPUTER-COMPATIBLE TAPE ASSIGNED TO LOGICAL UNIT 3.
C
C
C ORIGINS & WINDOWS ARE SPECIFIED BY SCANNER, GEOGRAPHIC,
C OR UTM COORDINATES. IN ORDER TO MINIMIZE TAPE ACCESSES,
C THE FIRST ORIGIN SHOULD HAVE THE LOWEST LINE NUMBERS, AND
C EACH SUCCEEDING ORIGIN PROGRESSIVELY HIGHER SCAN LINE
C NUMBERS.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C NONE.
C
C
C EXTERNAL REFERENCES
C -----
C
C NVIATO & NAME 'VIA' 'TO' ROUTINES
C VIATO & CALL 'VIA' 'TO' ROUTINES
C VIA TO
C EXTERNAL PIC000. PICXQT
C
C
C EXCEPTIONS
C -----
C
C 1. IF CONTROL HAS NOT BEEN EXECUTED IN THE CURRENT RUN PRIOR TO
C PICTAB AND SATISFACTORILY ADJUSTED A CONTROL NETWORK FOR THE
C SCENE TO BE PROCESSED BY PICTAB, THEN PICTAB WILL USE
C NOMINAL REGISTRATION PARAMETERS AND NOMINAL SCENE CENTER.
C
C 2. IF PICXQT DOES NOT CALL NVIATO TO CHANGE THE 'VIA' AND/OR 'TO'
C ROUTINES, THEN PICTAB WILL CALL TO PICXQT IN AN ENDLESS LOOP!
C
C
C GLOBAL DECLARATIONS
C -----

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICTAB
002

C

INCLUDE KONXQT.LIST	COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMLOG.LIST	COMMON LOG FILE BUFFER, I/O PKT. POINTERS
INCLUDE KOHLU3.LIST	COMMON POINTERS/FLAGS FOR UNIT 3
INCLUDE KOHLU5.LIST	COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
INCLUDE KONIHW.LIST	COMMON INPUT WINDOW PACKETS
INCLUDE KOMOHV.LIST	COMMON OUTPUT WINDOW PACKETS
INCLUDE KONNER.LIST	COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST	COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST	COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KONIRT.LIST	COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
INCLUDE KOMALT.LIST	COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS
INCLUDE KOMSYM.LIST	COMMON SYMBOL TABLE
INCLUDE KOMKS.LIST	COMMON COLOR SCREEN PARAMETERS
INCLUDE KOMTBL.LIST	COMMON MULTI-PURPOSE TABLE

C

C

C PROCEDURE

C -----

C

C

```
CALL NVIATO( PIC000.PICXQT)      * FIRST CALL IS VIA PIC000 TO PICXQT
100 CONTINUE
    CALL VIATO
GO TO 100
END      * (STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE)
```


PROGRAM PICTAB/VIRTUAL

HISTORY

E H SCHLOSSER	LEC	08/02/74	ORIGINAL CODE
E H SCHLOSSER	LEC	11/06/79	MAP.FZ(N): NO 'N' IN DEMAND

METHOD

CONSTRUCT MAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT XQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE MAP & XQT COMMANDS TO TEMPORARY FILE 20.
ADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 8-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS	3	FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IOWS	3	INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS	3	TERMINATE PROGRAM EXECUTION
DAM.PICTAB-MAP	3	SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAOPT	3	STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS BASO-D & PREP-D..

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:
(2 = REAL TIME
(3 = LOW EXEC
(4 = DEMAND
(5 = DEADLINE BATCH
(6 = BATCH

(XQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER A5 IN

(MASTER BIT NOTATION.

LOCAL DECLARATIONS

```

      AXRS
S(00) . D-BANK
SSSH   FORM          6.6.6.18
      11111222223333334444555555666666777777888888999999
LABSDF  SSSH          050.1.'F'.0 . LABEL. 1 WD. FORTRAN. FIELDATA
      *SOFF*
MAPSDF  SSSH          000.9.0.0 . DATA. 9 WDS. . FIELDATA
MAPINO  *BXQTS: MAP.FZN DAM.PICTAB-MAP.PICTAB . :BXQTS*
ADDSDF  SSSH          000.9.0.0
ADDINO  *BXQTS: ADD DAM.SYS-MAPOPT . :BXQTS*
XQTSDF  SSSH          000.9.0.0
XQTING  *BXQTS: XQT.1 . PICTAB . :BXQTS*
EOFSDF  -             0 . END-OF-FILE STOP WORD
PF      FORM          12.6.18
CSFASO  *BASO.T 20. . .
CSFADD  *BADD 20. . .
SAVREG  RES          1
IOPKT   I900         *20*.NS 33.LABSDF.'0' 0
  
```

PROCEDURE

```

S(01) . I-BANK
PICTAB* LA,U         A0. . . . . A0 := . . .
      TNE.U         A4.4 . . . . . SKIP NEXT INST IF A4<>4 (NOT DEMAND)
      SA.S2        A0.MAPINO*2 . DEMAND! BLANK OUT N OPTION
      LA          A0.(CSFASO) . ADDRESS OF BASO IMAGE
      ER          CSFS . DO IT
      SA          A0.SAVREG . STORE &
      PSRINT      (PF 2.1.SAVREG) . PRINT BASO STATUS
      GETOPT      . LOAD OPT LTRS INTO A2.A3.A4
PUTOPT  DS         A2.XQTING*2 . STORE OPTION LETTERS INTO BXQT IMAGE
      SA         A4.XQTING*4 . (3 WORDS -- MAX 18 OPT LETTERS)
WRITE  LA         A0.(IOPKT) . ADDRESS OF I/O PACKET
      ER         IOWS . WRITE SDF IMAGES TO 20.
ADD    LA         A0.(CSFADD) . ADDRESS OF BADD IMAGE
      ER         CSFS . DO IT
      ER         EXITS
      END        PICTAB
  
```

. PICTAB OVERLAY STRUCTURE

.

. HISTORY

.

E H SCHLOSSER	LEC	03/19/76	ORIGINAL CODE
E H SCHLOSSER	LEC	07/14/78	CHANGE OVERLAYS TO REDUCE THRASHING
E H SCHLOSSER	LEC	01/19/79	MACRO COMMANDS & TIME COMMAND
J C CRISP	LEC	10/24/79	PEEK, POKE, COLOR, INTENSITY, PICTURE
J C CRISP	LEMSCO	01/10/80	IF, FI & OPTIMIZE OVERLAYS
J C CRISP	LEMSCO	05/16/80	CROSSTAB, CHANGE DISTAB TO KNOTAB

. LIB DAM.

SEG S-MAIN

IN DAM.PICTAB/ . MAIN PROGRAM
IN DAM.NVIATO . NAME/CALL 'VIA' AND 'TO' SUBROUTINES
IN DAM.NULSUB . DO NOTHING
IN DAM.SYS-BLOCK . BLOCK DATA SUBROUTINE

. UTILITIES FOR MAIN -----

SEG S-FLINFO*(S-MAIN)

IN DAM.FLINFO . GET FILE DESCRIPTIVE INFORMATION

SEG S-R3TASCRTB*(S-MAIN)

IN DAM.R3TREC . READ ONE RECORD FROM TAPE (UNIT 3)
IN DAM.CSTVAS . CHARACTER STRING FOR ASCII
IN DAM.ISRTBA . INTEGER BUBBLE SORT ASCENDING

. MONITOR FOR PHASE 0.1.2.9 COMMANDS -----

SEG S-PIC0129*(S-FLINFO,S-R3TASCRTB)

IN DAM.PIC000 . CALL USER-SPECIFIED PHASE 0 ROUTINE
IN DAM.PIC129 . CALL PREVIOUSLY NAMED PHASE 1/2/9 'TO' ROUTINE
IN DAM.NTABS/DAM . DAM UNIT * TABLE GOES IN SAME SEG W/ FORTRAN I/O

. UTILITIES FOR PHASE 0.1.2.9 COMMANDS -----

SEG S-READS*(S-PIC0129)

IN DAM.READS . 'READ' INTO UNIT 5 BUFFER
IN DAM.GETS . 'GET' FREE-FORMAT FIELD FROM UNIT 5 BUFFER
IN DAM.WARNS . PROCESS WARNING DIAGNOSTIC FOR UNIT 5 FIELD
IN DAM.SPANS . ENABLE/DISABLE SPANNING FOR UNIT 5

SEG S-OPNCLPR*(S-PIC0129)

IN DAM.OPRPIC . OPEN ALT PRT FILES
IN DAM.CLOSPR . CLOSE ALT PRT FILES

SEG S-CALCROP*(S-PIC0129)

IN DAM.CALSYH . CALIBRATE SYMBOL TABLE
IN DAM.CALSPA . CALIBRATE PRINT/PLOT COEFFICIENTS FOR SPACING
IN DAM.CALWIN . CALIBRATE WINDOW
IN DAM.CALCHA . CALIBRATE CHANNEL POINTERS
IN DAM.CROPON . CROP SCANNER OUTPUT WINDOW TO FIT

. PHASE 0.1.2.9 COMMANDS (FORTRAN I/O ALLOWED) -----

SEQ S-XQTEX1*(S-READS,S-OPNCLPR,S-CALCROP)
IN DAM.PICXQT . PICTAB INITIALIZATION ROUTINE
IN DAM.PICEXI . PICTAB TERMINATION ROUTINE

SEQ S-PSTART*(S-XQTEX1)
IN DAM.PSTART . GENERAL INITIALIZATION ROUTINE

SEQ S-OPEN3*(S-XQTEX1)
IN DAM.OPEN3 . OPEN INPUT SCAN DATA FILE (UNIT 3)

SEQ S-OP3DSK*(S-OPEN3)
IN DAM.OP3DSK . OPEN INPUT -- DISK IN PX8DEF FMT (UNIT 3)

SEQ S-OP3BIP*(S-OPEN3)
IN DAM.OP3BIP . OPEN MSS DATA IN BIP FMT (UNIT 3)

SEQ S-OP3MDP*(S-OPEN3)
IN DAM.OP3MDP . OPEN MSS DATA IN MDP FMT (UNIT 3)

SEQ S-03TORHDR*(S-OP3MDP)
IN DAM.03TOR . MDP FMT TAPE DIRECTORY RECORD (UNIT 3)
IN DAM.03HDR . MDP FMT HEADER RECORD (UNIT 3)

SEQ S-03ANOT*(S-OP3MDP)
IN DAM.03ANOT . MDP FMT ANNOTATION RECORDS (UNIT 3)

SEQ S-03SZAM*(S-03ANOT)
IN DAM.03SZAM . SIZE AND INPUT WINDOW FOR AM TAPES

SEQ S-03SZPM*(S-03ANOT)
IN DAM.03SZPM . SIZE AND INPUT WINDOW FOR PM TAPES

SEQ S-03SZAR*(S-03ANOT)
IN DAM.03SZAR . SIZE AND INPUT WINDOW FOR AR TAPES

SEQ S-03SZPR*(S-03ANOT)
IN DAM.03SZPR . SIZE AND INPUT WINDOW FOR PR TAPES

SEQ S-03ANCL*(S-OP3MDP)
IN DAM.03ANCL . MDP FMT ANCILLARY RECORDS (UNIT 3)

SEQ S-LDREG00*(S-XQTEX1)
IN DAM.LDREG00 . LOAD REGISTRATION PARAMETERS FROM UNIT 0

SEQ S-CLOSTOP*(S-XQTEX1)
IN DAM.CLOSE3 . CLOSE INPUT SCAN DATA FILE (UNIT 3)
IN DAM.PSTOP . GENERAL TERMINATION ROUTINE

SEG S-HELP*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KHDCLE . CLEAR WARNINGS/ERRORS
IN DAM.KHDEXP . EXPLAIN PROGRAM/COMMAND
IN DAM.KHDFI . END IF...FI BLOCK
IN DAM.KHDIF . BEGIN IF...FI BLOCK
IN DAM.KHDNEW . PRINT NEWS
IN DAM.KHDNEW . CONDITIONALLY PERFORM NEXT COMMAND
IN DAM.KHDOFF . TURN OFF MODE SWITCH(ES)
IN DAM.KHDON . TURN ON MODE SWITCH(ES)
IN DAM.KHDPEE . PEEK
IN DAM.KHDPOK . POKE
IN DAM.KHDREN . RENUMBER (GET/CHECK NEW WINDOW SEQUENCE NUMBER)
IN DAM.KHDTIM . PRINT CLOCK TIME & CHARGE TIME

SEG S-GEOMETRY*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KHDALI . ALIGN COORDINATE SYSTEMS
IN DAM.KHDMER . GET/CHECK TRANSVERSE MERCATOR CENT MERIDIAN
IN DAM.KHDTIC . GET/CHECK TICK INTERVALS
IN DAM.KHDWIN . GET/CHECK WINDOW ENVELOPE/VERTICES
IN DAM.KHDZON . GET/CHECK UTM PROJECTION ZONE

SEG S-SPECS-9*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KHDCHA . GET/CHECK RAW/TRANSFORMED SCANNER CHANNEL(S)
IN DAM.KHDHEA . GET/CHECK PAGE HEADING(S)
IN DAM.KHDNAM . GET/CHECK TRANSFORMATION/MATERIAL NAME
IN DAM.KHDORI . GET/CHECK WINDOW ORIGIN
IN DAM.KHDRAD . GET/CHECK RADIANCE LIMITS
IN DAM.KHDSPA . GET/CHECK WINDOW SPACING
IN DAM.PICDI9 . DISPLAY RADIANCE/GRADIENT/CLASS (PHASE 9)
IN DAM.PICLI9 . LIST RADIANCE/GRADIENT/CLASS (PHASE 9)
IN DAM.PICPA9 . PARTITION FACTOR SPACE (PHASE 9)
IN DAM.PICPI9 . PICTURE RADIANCE/GRADIENT/CLASS (PHASE 9)
IN DAM.PICPR9 . PROFILE RADIANCE (PHASE 9)
IN DAM.PICTO9 . TOTAL TABULATIONS (PHASE 9)

SEG S-NISC*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KHDCOL . GET/CHECK COLORS
IN DAM.KHDCOP . GET/CHECK NUMBER OF OUTPUT COPIES
IN DAM.KHDCRO . CROSSTABULATE COLOR & INTENSITY
IN DAM.DISHIS . HISTOGRAM PREVIOUSLY DISPLAYED DATA (PHASE 0)
IN DAM.KHDINT . GET/CHECK INTENSITY
IN DAM.KHDLIN . GET/CHECK LINEAR TRANSFORMATION WEIGHTS/GAIN/BIAS
IN DAM.KHDPAG . SKIP TO TOP OF NEXT PAGE
IN DAM.KHDPRI . GET/CHECK PRINTER SPECIFICATIONS
IN DAM.KHDPOL . GET/CHECK POLAR TRANSFORMATION GAIN/BIAS
IN DAM.KHDSHA . GET/CHECK SHARPENING FILTER COEFFICIENTS
IN DAM.KHDSYM . GET/CHECK SYMBOLS
IN DAM.KHDTAB . TABULATE BY RADIANCE/SYMBOL/COLOR/INTENSITY

SEG S-EXEC*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KHDXXX . MACRO COMMANDS
IN DAM.KHDGAD . DYNAMIC GADD
IN DAM.KHDGAS . DYNAMIC GASO
IN DAM.KHDGGR . DYNAMIC GBRKPT
IN DAM.KHDGFR . DYNAMIC GFREE

IN DAH.KND0LO . DYNAMIC 8LO0

SEG S-DISLSPIC*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAH.PICDIS . DISPLAY RADIANCE/GRADIENT/CLASS (PHASE 0)
IN DAH.PICLIS . LIST RADIANCE/GRADIENT/CLASS (PHASE 0)
IN DAH.PICPIC . PICTURE RADIANCE/GRADIENT/CLASS (PHASE 0)

SEG S-FACROT*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAH.PICFAC . FACTOR CHANNELS (PHASE 0)
IN DAH.PICFA9 . FACTOR CHANNELS (PHASE 9)
IN DAH.PICROT . ROTATE FACTOR STRUCTURE/COEFFICIENTS (PHASE 0)

SEG S-PICPAR*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAH.PICPAR . PARTITION FACTOR SPACE (PHASE 0)

SEG S-PICPRO*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAH.PICPRO . PROFILE RADIANCE (PHASE 0)

SEG S-PICTOT*. (S-READS.S-OPNCLPR.S-CALCROP)
IN DAH.PICTOT . TOTAL TABULATIONS (PHASE 0)

. MONITOR FOR PHASE 3.4.5 COMMANDS -----

SEG S-PIC345*. S-PIC0129
IN DAH.PIC345 . CALL PREVIOUSLY NAMED PHASE 3/4/5 'TO' ROUTINE

. UTILITIES FOR PHASE 3.4.5 COMMANDS -----

SEG S-RD3HDP*. (S-PIC345)
IN DAH.RD3BIL . READ HSS DATA IN BIL FORMAT (UNIT 3)
IN DAH.RD3BSQ . READ HSS DATA IN BSQ FORMAT (UNIT 3)

SEG S-RD3BIP*. (S-PIC345)
IN DAH.RD3BIP . HSS DATA IN BIP FORMAT (UNIT 3)

SEG S-RD3DSKNUL*. (S-PIC345)
IN DAH.RD3DSK . DATA ON DISK IN PX8DEF FORMAT
IN DAH.RD3NUL . SYNTHETIC DATA -- NO UNIT 3

. PHASE 3.4.5 COMMANDS (NO FORTRAN I/O) -----

SEG S-PICD13*. (S-RD3HDP.S-RD3BIP.S-RD3DSKNUL)
IN DAH.PICD13 . DISPLAY RADIANCE
SEG S-PICD14*. (S-RD3HDP.S-RD3BIP.S-RD3DSKNUL)
IN DAH.PICD14 . DISPLAY GRADIENT/LAPLACIAN/VARIANCE
SEG S-PICD15*. (S-RD3HDP.S-RD3BIP.S-RD3DSKNUL)
IN DAH.PICD15 . DISPLAY CLASS

SEG S-PICFA3*. (S-RD3HDP.S-RD3BIP.S-RD3DSKNUL)
IN DAH.PICFA3 . FACTOR CHANNELS

SEG S-PICL13*. (S-RD3HDP.S-RD3BIP.S-RD3DSKNUL)
IN DAH.PICL13 . LIST RADIANCE

SEO S-PICL14*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICL14 . LIST GRADIENT/LAPLACIAN/VARIANCE
SEO S-PICL15*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICL15 . LIST CLASS

SEO S-PICPA3*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICPA3 . PARTITION BY DENSITY
SEO S-PICPA4*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICPA4 . PARTITION BY GRADIENT/LAPLACIAN/VARIANCE

SEO S-PICPI3*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICPI3 . PICTURE RADIANCE
SEO S-PICPI4*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICPI4 . PICTURE GRADIENT/LAPLACIAN/VARIANCE
SEO S-PICPI5*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICPI5 . PICTURE CLASS

SEO S-PICPR3*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICPR3 . PROFILE RADIANCE

SEO S-PICT03*(S-RD3HDP,S-RD3BIP,S-RD3DSKNUL)
IN DAM.PICT03 . TOTAL TABULATIONS

. MONITOR FOR PHASE 6.7.8 COMMANDS -----

SEO S-PIC670*,S-PIC0129
IN DAM.PIC670 . CALL PREVIOUSLY NAMED PHASE 6/7/8 'TO' ROUTINE

. PHASE 6.7.8 COMMANDS (NO FORTRAN I/O) -----

SEO S-PICPA6*(S-PIC670)
IN DAM.PICPA6

**DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**PICTAB-NAP/VIRTUAL
001**

IN DAN.PICTAB/VIRTUAL


```

C
  INTEGER KOND          & FIRST 3 CHARS OF USER COMMAND (BLANK AFTER DONE)
  INTEGER LSSTAT       & READS STATUS ('EOF' MEANS END-OF-FILE)
  INTEGER KASE         & MODIFIED I-C-E OF FIRST CHAR OF COMMAND

C
C
C PROCEDURE
C -----
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
  CALL TRACE
  CALL NANSUB          & CALL TO NULSUB DOES NOTHING

C
C
C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C
  KOND=' NUL'         & IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
  IF(INSBATCH.NE.0) CALL READS(LSSTAT, ' ') & FILL BUFFER, BLANK CUE MSG
  IF(INSBATCH.EQ.0) CALL READS(LSSTAT, NULCST) & FILL BUFFER, NO CUE MSG
  IF(LSSTAT.NE.' ') KOND='EOF5'
  IF(KOND.NE.'EOF5') CALL GETSAL(KOND,(3), NULCST) & GET 3 ALPHA CHARS

C
C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C
  KASE=ICE(KOND)-ICE('A')+1      & A TO Z = 1 TO 26

C
C
C CASE STATEMENT ON MODIFIED I-C-E OF COMMAND'S FIRST CHARACTER
C
  IF((KASE.LT.1).OR.(KASE.GT.26)) KASE=27      & NOT ALPHA
  GO TO 1
  0 401,402,403,404,405,406,407,408,409,410.
  1 411,412,413,414,415,416,417,418,419,420.
  2 421,422,423,424,425,426,427)
  3 .KASE

C
C
C DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOND TO BLANK
C
  401 CONTINUE &... A
    IF(KOND.EQ.'AL') CALL KMDAL(KOND)          & ALIGN
    GO TO 000

C
  402 CONTINUE &... B
    GO TO 000

C
  403 CONTINUE &... C
    IF(KOND.EQ.'CHA') CALL KMDCHA(KOND)        & CHANNEL
    IF(KOND.EQ.'CLE') CALL KMDCLE(KOND)        & CLEAR
    IF(KOND.EQ.'COL') CALL KMDCOL(KOND)        & COLOR
    IF(KOND.EQ.'COP') CALL KMDCOP(KOND)        & COPIES
    IF(KOND.EQ.'CRO') CALL KMDCRO(KOND)        & CROSSTAB
    GO TO 000
  
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PIC000
003

C
404 CONTINUE 3*** D
IF(KOMD.EQ.'DIS') CALL PICDIS(KOMD) & DISPLAY
GO TO 800

C
405 CONTINUE 3*** E
IF(KOMD.EQ.'EOP5') CALL PICEXI(KOMD) & END-OF-FILE CAUSES EXIT
IF(KOMD.EQ.'EXI') CALL PICEXI(KOMD) & EXIT
IF(KOMD.EQ.'EXP') CALL KMDEXP(KOMD) & EXPLAIN
GO TO 800

C
406 CONTINUE 3*** F
IF(KOMD.EQ.'FAC') CALL PICFAC(KOMD) & FACTOR
IF(KOMD.EQ.'FI ') CALL KMDFI (KOMD) & FI (ENDIF)
GO TO 800

C
407 CONTINUE 3*** G
GO TO 800

C
408 CONTINUE 3*** H
IF(KOMD.EQ.'HEA') CALL KMHHEA(KOMD) & HEADING
IF(KOMD.EQ.'HIS') CALL DISHIS(KOMD) & HISTOGRAM
GO TO 800

C
409 CONTINUE 3*** I
IF(KOMD.EQ.'IF ') CALL KMDIF (KOMD) & IF
IF(KOMD.EQ.'INT') CALL KMDINT(KOMD) & INTENSITY
GO TO 800

C
410 CONTINUE 3*** J
411 CONTINUE 3*** K
GO TO 800

C
412 CONTINUE 3*** L
IF(KOMD.EQ.'LIN') CALL KMDLIN(KOMD) & LINEAR
IF(KOMD.EQ.'LIS') CALL PICLIS(KOMD) & LIST
GO TO 800

C
413 CONTINUE 3*** M
IF(KOMD.EQ.'MER') CALL KMDMER(KOMD) & MERIDIAN
GO TO 800

C
414 CONTINUE 3*** N
IF(KOMD.EQ.'NAM') CALL KMDNAM(KOMD) & NAME
IF(KOMD.EQ.'NEW') CALL KMDNEW(KOMD) & NEWS
IF(KOMD.EQ.'NEX') CALL KMDNEX(KOMD) & NEXT
GO TO 800

C
415 CONTINUE 3*** O
IF(KOMD.EQ.'OFF') CALL KMDOFF(KOMD) & OFF
IF(KOMD.EQ.'ON ') CALL KMDON (KOMD) & ON
IF(KOMD.EQ.'ORI') CALL KMDOR(KOMD) & ORIGIN
GO TO 800

C
416 CONTINUE 3*** P
IF(KOMD.EQ.'PAG') CALL KMDPAG(KOMD) & PAGE

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PIC000
004

	IF(KOMD.EQ.'PAR') CALL PICPAR(KOMD)	▣ PARTITION
	IF(KOMD.EQ.'PEE') CALL KMDPEE(KOMD)	▣ PEEK
	IF(KOMD.EQ.'PIC') CALL PICPIC(KOMD)	▣ PICTURE
	IF(KOMD.EQ.'POK') CALL KMDPOK(KOMD)	▣ POKE
	IF(KOMD.EQ.'POL') CALL KMDPOL(KOMD)	▣ POLAR
	IF(KOMD.EQ.'PRI') CALL KMDPRI(KOMD)	▣ PRINTER
	IF(KOMD.EQ.'PRO') CALL PICPRO(KOMD)	▣ PROFILE
	GO TO 800	
C	417 CONTINUE 3*** Q	
	GO TO 800	
C	418 CONTINUE 3*** R	
	IF(KOMD.EQ.'RAD') CALL KMDRAD(KOMD)	▣ RADIANCE
	IF(KOMD.EQ.'REN') CALL KMDREN(KOMD)	▣ RENUMBER
	IF(KOMD.EQ.'ROT') CALL PICROT(KOMD)	▣ ROTATE
	GO TO 800	
C	419 CONTINUE 3*** S	
	IF(KOMD.EQ.'SHA') CALL KMDSHA(KOMD)	▣ SHARPENING
	IF(KOMD.EQ.'SPA') CALL KMDSPA(KOMD)	▣ SPACING
	IF(KOMD.EQ.'SYM') CALL KMDSYM(KOMD)	▣ SYMBOLS
	GO TO 800	
C	420 CONTINUE 3*** T	
	IF(KOMD.EQ.'TAB') CALL KMDTAB(KOMD)	▣ TABULATE
	IF(KOMD.EQ.'TIC') CALL KMDTIC(KOMD)	▣ TICKS
	IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD)	▣ TIME
	IF(KOMD.EQ.'TOT') CALL PICTOT(KOMD)	▣ TOTAL
	GO TO 800	
C	421 CONTINUE 3*** U	
	422 CONTINUE 3*** V	
	GO TO 800	
C	423 CONTINUE 3*** W	
	IF(KOMD.EQ.'WIN') CALL KMDWIN(KOMD)	▣ WINDOW
	GO TO 800	
C	424 CONTINUE 3*** X	
	425 CONTINUE 3*** Y	
	GO TO 800	
C	426 CONTINUE 3*** Z	
	IF(KOMD.EQ.'ZON') CALL KMDZON(KOMD)	▣ ZONE
	GO TO 800	
C	427 CONTINUE 3*** NOT ALPHABETIC -- **ONLY FOR DEBUGGING**	
	IF(KOMD.EQ.'3AD') CALL KMD3AD(KOMD)	▣ 3ADD
	IF(KOMD.EQ.'SAD') CALL KMD3AD(KOMD)	▣ SADD
	IF(KOMD.EQ.'SAS') CALL KMD3AS(KOMD)	▣ SASO
	IF(KOMD.EQ.'SBR') CALL KMD3BR(KOMD)	▣ 3BRKPT
	IF(KOMD.EQ.'SFR') CALL KMD3FR(KOMD)	▣ 3FREE
	IF(KOMD.EQ.'SLO') CALL KMD3LO(KOMD)	▣ 3SLOO
C		
C		

ORIGINAL PAGE 10
OF FOUR QUAL

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PIC000
005

```
C IF COMMAND WAS NOT FOUND. TRY MACRO-COMMAND
C
000 IF(KOMD.NE.' ') KOMD='PIC-'      & 1ST 3 CHARS OF PROG NAME PLUS '--'
      IF(KOMD.NE.' ') CALL KMDXXX(KOMD)  & MACRO COMMAND HANDLER
C
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
      IF(KOMD.NE.' ') CALL WARNS( 'INVALID COMMAND --')
C
C
C FORCE ALL FORTRAN I/O ROUTINES INTO SAME SEG AS PIC000 (NEVER PERFORMED)
C
      IF(KOMD.EQ.'JUNK') READ(895,895) KOMD
      895 FORMAT(IX)
C
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
      RETURN
      END
```

SUBROUTINE PIC129: 8 CALL PHASE 1/2/9 SUBROUTINES FOR PICTAB
1 NAMSUB) 8 NAME OF SUBROUTINE TO CALL

C -----
C
C (E H SCHLOSSER)
C
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
C CALL TRACE
C CALL NAMSUB
C
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
C RETURN
C END

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PIC348
001

```
      SUBROUTINE PIC348( 8 CALL PHASE 3/4/6 SUBROUTINES FOR PICTAB  
      I NANSUB)      8 NAME OF SUBROUTINE TO CALL  
      -----  
C  
C (E H SCHLOSSER)  
C  
C  
C CALL PREVIOUSLY NAMED SUBROUTINE  
C  
      CALL TRACE  
      CALL NANSUB  
C  
C  
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY  
C  
      RETURN  
      END
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PIC670
001

SUBROUTINE PIC670/ 0 CALL PHASE 6/7/8 SUBROUTINES FOR PICTAB
1 NANSUB) 0 NAME OF SUBROUTINE TO CALL

C
C
C (E W SCHLOSSER)
C
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
C CALL TRACE
C CALL NANSUB
C
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
C
C RETURN
C END

- C
 C 2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE DISPLAY.
 C
 C 3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS
 C

CONDITION -----	ACTION -----	DIAGNOSTIC -----
C PROCESSING DEFAULT COMMANDS C (NMNDOW=0)	NONE	WARNING
C KLSTYP IN COMMON UNDEFINED	KLSTYP:='RAD'	NONE
C KLSTYP SPECIFICATION MISSING	USE COMMON KLSTYP	NONE
C KLSTYP SPECIFICATION INVALID	NONE	WARNING
C EXTRA SPECIFICATION	NONE	WARNING
C LIMIT CHANNEL VALUE RANGE IS NULL C (LCVLOI>LCVHI)	NONE	WARNING
C SPACING > 0.5 OR < 1.5	DON'T GENERATE TICKS	NONE
C NUMBER CONTROL POINTS < 8	DON'T LIST TICKS	NONE
C DEMAND RUN & OVERPRINTED SYMBOLS	NO OVERPRINTING IN PICD13	NOTE
C DATA/CHECKOUT MODE	'TO' ROUTINE IS NULSUB	NONE
C WARNING(S) OR FATAL ERROR(S)	'TO' ROUTINE IS PICD13	NONE

C GLOBAL DECLARATIONS
 C -----
 C

INCLUDE KOMXQT.LIST	8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST	8 COMMON CLASSIFICATION SPECTRAL LIMITS
INCLUDE KOMFIT.LIST	8 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMTBL.LIST	8 COMMON TICK/FREQ/FACTOR TABLE
INCLUDE KONSYM.LIST	8 COMMON SYMBOL TABLE
INCLUDE WINDEF.LIST	8 DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMOHM.LIST	8 COMMON OUTPUT WINDOW PACKETS
INCLUDE NULCST.LIST	8 DEFINE NULL CHARACTER STRING

C
 C LOCAL DECLARATIONS
 C -----
 C

INTEGER KLSAVE	8 SAVE LOCATION FOR KLSTYP
INTEGER INTEMP	8 TEMPORARY
INTEGER LUNTC	8 LOGICAL UNIT NUMBER TO LIST TICK COORDS ON
INTEGER NPRLIN,NPRCOL	8 NUMBER OF PRINT LINES/COLUMNS IN DISPLAY
INTEGER LUNALT	8 LOGICAL UNIT NUMBER TO SPOOL PRINTER OUTPUT
INTEGER NPRCIF	8 NUMBER OF PRINT COLUMNS IN ONE FILE

C
 C PROCEDURE
 C -----
 C

CALL TRACE

C
 C GET DISPLAY TYPE
 C

KTBLTY=' NUL ' 8 MARK OLD FREQ TABLE AS DESTROYED

```

IF(NWNDOW.EQ.0) CALL MDWARN( 'INVALID DEFAULT COMMAND')
KLSAVE=KLSTYP      & SAVE PREVIOUS DISPLAY TYPE
IF((KLSTYP.NE.'GRA').AND.
& (KLSTYP.NE.'LAP').AND.
& (KLSTYP.NE.'VAR').AND.
& (KLSTYP.NE.'CLA'))
& KLSTYP='RAD'      & IF UNDEFINED ...
CALL GETSKH(KLSTYP,(3), NULCST) & ... THEN MAKE IT RADIANCE1 ...
& ... UNLESS SPECIFIED BY DISPLAY) CMD

C
C CHECK DISPLAY TYPE
C
IF(KLSTYP.NE.'RAD') GO TO 240      & RADIANCE1?
CALL NVIATO( PIC345.PICD13) & NEXT CALL IS TO PICD13
GO TO 300
240 IF((KLSTYP.NE.'GRA').AND.      & GRA(DIENT)?
& (KLSTYP.NE.'LAP').AND.        & LAP(LACIAN)?
& (KLSTYP.NE.'VAR')) GO TO 250  & VAR(IANCE)?
CALL NVIATO( PIC345.PICD14) & NEXT CALL IS TO PICD14
GO TO 300
250 IF(KLSTYP.NE.'CLA') GO TO 280  & CLA(SS)?
CALL NVIATO( PIC345.PICD15) & NEXT CALL IS TO PICD15
GO TO 300
280 CALL WARN5( 'BAD DISPLAY TYPE --')
KLSTYP=KLSAVE      & RESTORE PREVIOUS DISPLAY TYPE

C
C DRAIN SPECS FOR CURRENT COMMAND
C
300 CALL GETSIN(INTERP. +1,-1,'EXTRA DISPLAY SPECIFICATION --')

C
C CHECK RADIANCE LIMITS
C
IF(LCVLO1.GT.LCVHI1) CALL MDWARN( 'NO RADIANCE LIMITS')
IF(MDATA.C.NE.0) GO TO 900      & DATA/CHECKOUT MODE

C
C CALIBRATE CHANNELS/SYMBOLS/SPACING/WINDOW
C
CALL CALCHA
CALL CALSYM
CALL CALSPA
CALL CALWIN( 0.)

C
C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
IF(NDTOTL.NE.0) GO TO 900
IF(NWNDOW.LT.0) CALL OPRPIC      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
NWNDOW=1ABS(NWNDOW)
NPAGE=0

C
C CROP OUTPUT WINDOW TO FIT INPUT WINDOW & ALT PRINT FILE(S)
C

```


DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICDIS
004

```
      NPRCIF=KPAGE-8      & NUMBER OF PRINT COLUMNS PER FILE EXCLUDING
C      & LEFT MARGIN AND 2 HEAT LINE CHARS
      CALL CROPWH( 1000,MALTM*NPRCIF)
      IF (INDTOTL.NE.0) GO TO 900
C
C
C GENERATE TICKS (LIST ON UNIT 10 IF REGISTRATION ACCURATE)
C
      LUNTIC=0
      IF (INCLTPT.GT.5) LUNTIC=10      & LIST TICKS ON UNIT 10
      IF ((MSAOWH(WLIN,WSP100).LT.50).OR.
& (MSAOWH(WLIN,WSP100).GT.150)) LUNTIC=-1      & NO TICKS
      IF ((MSAOWH(WSAM,WSP100).LT.50).OR.
& (MSAOWH(WSAM,WSP100).GT.150)) LUNTIC=-1      & NO TICKS
      CALL GENTIC( LUNTIC)
C
C
C COMPUTE SIZE OF PRINT WINDOW
C
      NPRLIN=IFIX(PPDOWH(WLIN,WMAX))-IFIX(PPDOWH(WLIN,WMIN))+1
      NPRCOL=IFIX(PPDOWH(WCOL,WMAX))-IFIX(PPDOWH(WCOL,WMIN))+1
C
C
C CHECK FOR DIAGNOSTICS
C
      IF (INDTOTL.NE.0) GO TO 900
      IF ((MIBATCH.EQ.0).AND.(NCISYM.NE.1)) CALL MDNOTE(
& 'SYMBOLS NOT OVERPRINTED ON DEMAND TERMINAL')
C
C
C PRINT WINDOW HEADING FOR UNIT 6
C
      WRITE(6,415) NWNDOW,MTERAL
      415 FORMAT(' WINDOW NUMBER ',J3,6X,'DISPLAY',6X,4A8)
      CALL IDLU3( 6)
      CALL IDERT( 6)
      CALL IDCPIG( 6)
C
C
C PRINT WINDOW HEADING FOR ALTERNATE PRINT FILE(S)
C
      LUNALT=10
      DO 460 N=1,NPRCOL,NPRCIF
      CALL MDUNIT( 4,LUNALT)
      WRITE(LUNALT,415) NWNDOW,MTERAL
      CALL IDLU3( LUNALT)
      CALL IDERT( LUNALT)
      CALL IDCPIG( LUNALT)
      LUNALT=LUNALT+1
      460 CONTINUE
C
C
C ANY DIAGNOSTICS???
C
      900 IF (INDTOTL.EQ.0) GO TO 990
      IF (INDATAC.NE.0) CALL NVIATO( PIC000,NULSUB)      & DATA/CHECKOUT
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICD18
005

IF(MDATAC.EQ.0) CALL NVIAT01 (PIC129.PICD19)

C
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KOND=1
RETURN
END

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICD13
002

C	'BADR'	'7'	NO DATA (':')	NONE	PRINT LINE
C	'BADF'	N/A	N/A	FATAL	RETURN
C	'OFL'	N/A	N/N	FATAL	RETURN

C
C
C GLOBAL DECLARATIONS
C -----
C

INCLUDE KONXQT.LIST	% COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONKLS.LIST	% COMMON CLASSIFICATION INFO
INCLUDE KOSYH.LIST	% COMMON SYMBOL TABLE
INCLUDE WINDEF.LIST	% DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KONOWH.LIST	% COMMON OUTPUT WINDOW PACKETS
INCLUDE KOHTBL.LIST	% COMMON TICK/FREQ TABLE
INCLUDE PICDEF.LIST	% DEFINE PICTAB PARAMETERS
INCLUDE PXDEF.LIST	% DEFINE PIXEL BUFFER STRUCTURE
INCLUDE MAXINT.LIST	% MAXIMUM INTEGER VALUE

C
C
C LOCAL DECLARATIONS
C -----
C

PARAMETER NKBUFS=6 % # OF MBS PIXEL BUFFERS IN ARRAY
 INTEGERS IN MSA BUF = #INTS PREAMBLE * (PBINS*3)/4 * (EXTRA BYTES*3)/4
 PARAMETER NMIXBF = (PXBIN-1) * (354*3)/4 * (18*3)/4

MBS PRT BUF=MBS PREAM*(MAX FILES*(MAX COLS/PG-NEATLN COLS-MARG COLS))*N
 PARAMETER NMIPBF=(PXBIN-1)*(HALTNI * (KPAQNI- 2 - 4)) * 4

PARAMETER NFROCH=6 % # MBS PER CHANNEL IN LOCAL FREQ TABLE
 PARAMETER NFROSZ=128 % # CHANNELS IN LOCAL FREQ TABLE
 INTEGER MPXBUF(NMIXBF,NKBUFS) % ARRAY OF MBS PIXEL BUFFERS
 INTEGER (PRTBF(NMIPBF) % PRINT BUFFER
 INTEGER NFREQ(NFROSZ,NFROCH) % LOCAL FREQUENCY TABLE (SCOPE INCLUDES
 % INTERNAL ROUTINE RESSYM)

INTEGER IPLIN % PRINT LINE
 INTEGER IPCHIN,IPCHMAX % MINIMUM AND MAXIMUM PRINT COLUMN
 INTEGER IPLMIN,IPLMAX % MINIMUM AND MAXIMUM PRINT LINE
 REAL ADJLIN,ADJSAH % ADJUSTED LINE AND SAMPLE
 INTEGER ML100L,ML100M,ML100S % MSA LINE*100: LOW,HIGH,SPACING
 INTEGER MSALIN % MSA LINE NUMBER
 INTEGER MSASLO,MSASHI % LOW AND HIGH MSA SAMPLE
 INTEGER ISTAT % I/O STATUS
 INTEGER NTLCHR % NEAT LINE CHARACTER
 INTEGER MROLCS,MROPCS % LEFT & RIGHT MARGIN CHAR STRING
 INTEGER IPTIC,IPCTIC,JSYTIC % TICK PRINT LINE,COLUMN,SYMBOL
 INTEGER NPRLIN,NPRCOL % NUMBER OF PRINT LINES AND COLUMNS
 INTEGER LASTLN % LAST SCAN LINE READ

C
C
C PROCEDURE
C -----
C

CALL TRACE

C
C

C

```

IPLIN=IPLMIN
LASTLN=-MAXINT
DO 350 ML100=ML100L,ML100H,ML100S
  NTLCHR=':'
  MSALIN=ML100/100
  CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCMIN))
  MSASLO=ADJSAM
  CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCMAX))
  MSASHI=ADJSAM
  IF (MSALIN.NE.LASTLN) CALL GETRAD (MPXBUF,(NWIXBF),(NXBUFS),
    & ISTAT, MSALIN,MSASLO,MSASHI)
  LASTLN=MSALIN
  IF ((ISTAT.NE.'BADF').AND.(ISTAT.NE.'OFL')) GO TO 320
    CALL HDFATL (CBS4CS(ISTAT,1,4),
    & ' WHILE READING ON UNIT 3')
    CALL ERPRTA ('10',.2,6,
    & '*I/O ERROR - IGNORE OUTPUT*')
    GO TO 900
320 IF (ISTAT.EQ.'BADR') NTLCHR='?'
    CALL MSKPIX (MPXBUF(1,1), MPXBUF(1,1))
    IF ((MPXBUF(PXBINT,1).EQ.'BYT').OR.
    & (MPXBUF(PXBINT,1).EQ.'CHR').OR.
    & (MPXBUF(PXBINT,1).EQ.'INT').OR.
    & (MPXBUF(PXBINT,1).EQ.'NUL')) GO TO 330
    CALL HDFATL('INVALID BIN TYPE ',
    & CBS4CS(MPXBUF(PXBINT,1),1,3),' IN PICD13')
    GO TO 900
330 IF (MPXBUF(PXBINT,1).EQ.'BYT') CALL RESSYM (IPRTBF,(IPLIN),
    & (IPCMIN),(IPCMAX), MPXBUF,(NWIXBF),(NXBUFS),GETBYT)
    IF (MPXBUF(PXBINT,1).EQ.'CHR') CALL RESSYM (IPRTBF,(IPLIN),
    & (IPCMIN),(IPCMAX), MPXBUF,(NWIXBF),(NXBUFS),GETICE)
    IF (MPXBUF(PXBINT,1).EQ.'INT') CALL RESSYM (IPRTBF,(IPLIN),
    & (IPCMIN),(IPCMAX), MPXBUF,(NWIXBF),(NXBUFS),GETINT)
    IF (MPXBUF(PXBINT,1).EQ.'NUL') CALL RESSYM (IPRTBF,(IPLIN),
    & (IPCMIN),(IPCMAX), MPXBUF,(NWIXBF),(NXBUFS),GETNUL)
    CALL CST4IN (MRGLCS,(1),(6), MSALIN,4,'0')
    MRGRCS=MRGLCS
    IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
    & CALL PROVFI (6, MRGLCS,4,'.0,NTLCHR,'1*****',IPRTBF)
    CALL PROVFI (10, MRGLCS,4,MRGRCS,4,NTLCHR,'1000***',IPRTBF)
    IPLIN=IPLIN+1
350 CONTINUE
C
C
C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE AFTER LAST PRINT LINE
C
CALL SANSCL (IPRTBF,(IPLMAX+1),(IPCMIN),(IPCMAX))
IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
& CALL PROVFI (6, MRGLCS,4,'.0,NTLCHR,'1*****',IPRTBF)
CALL PROVFI (10, MRGLCS,4,MRGRCS,4,NTLCHR,'1000***',IPRTBF)
CALL PROVFI (10, MRGLCS,4,MRGRCS,4,NTLCHR,'31111***',IPRTBF)
C
C
C MOVE DATA FROM LOCAL FREQ TABLE TO COMMON TABLE (REPLACING TICKS)
C

```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICD13
007

C REAL ADJSAM & ADJUSTED SAMPLE
 INTEGER IPBIN & POINTER TO PRINT BIN
 INTEGER MSASAM & SAMPLE NUMBER
 INTEGER MS100L,MS100H,MS100S & MSA SAMPLE*100: LOW,HIGH,SPACING

C
C
C PROCEDURE

C INITIALIZE LOW AND HIGH SAMPLES AND SPACING

C
 CALL A4P (ADJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCHIN))
 MS100L=ADJSAM*100.
 CALL A4P (ADJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCHMAX))
 MS100H=ADJSAM*100.
 MS100S=MSAOWH/MSAM,MSPI00)

C
C
C SET PREAMBLE POINTERS AND BIN POINTER

C
 IPRTBF(PXRECN)=0
 IPRTBF(PXLINO)=IPLIN
 IPRTBF(PXCHAN)=0
 IPRTBF(PXQUAL)=0
 IPRTBF(PXBINT)='INT'
 IPRTBF(PXLBIN)=1
 IPRTBF(PXLCOL)=IPCHIN
 IPRTBF(PXHBIN)=IPCHMAX-IPCHIN+1
 IPRTBF(PXHCOL)=IPCHMAX
 IPRTBF(PXNCHN)=0
 IPRTBF(PXNODA)=0
 IPRTBF(PXLJOI)=0
 IPRTBF(PXHJOI)=0
 IPBIN=IPRTBF(PXLBIN)-1

C
C
C ENCODE SAMPLE NUMBER AND PUT COLON, STRING, AND COLON IN BUFFER

C
 DO 100 MS100=MS100L,MS100H,MS100S
 MSASAM=MS100/100
 CALL PUTCHR (IPRTBF(PXBINS+IPBIN),(1), ':')
 CALL CST4IN (IPRTBF(PXBINS+IPBIN),(2),4, MSASAM,4,'0')
 CALL PUTCHR (IPRTBF(PXBINS+IPBIN),(6), ':')
 IPBIN=IPBIN+1

100 CONTINUE

C
C
C RETURN

C
C
C
C
C
C
C

```
C      INTERNAL
C      SUBROUTINE RESSYM (      % RESAMPLE/SCREEN/COUNT FREQUENCY/SYMBOLIZE/
C                                % INSERT TICKS
C      O IPRTF.      % PRINT BUFFER
C      ( IPLIN.      % PRINT LINE
C      ( IPCHIN.     % MINIMUM PRINT COLUMN
C      ( IPCHAX.     % MAXIMUM PRINT COLUMN
C      .
C      I MPXBUF.     % ARRAY OF MSS PIXEL BUFFERS
C      ( NWIXBF.     % NUMBER OF WORDS IN ONE BUFFER
C      ( NXBUFS.     % NUMBER OF BUFFERS
C      I GETBIN)     % ROUTINE TO GET BIN VALUE--GETBYT.GETICE.GETINT.GETNUL
C
C      C
C      C METHOD
C      C
C      COMPUTE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER PREAMBLE.
C      FOR EACH SAMPLE. CHECK IF OUTSIDE OF INPUT WINDOW. CHECK IF OUT-
C      SIDE OF RADIANCE LIMITS. COUNT FREQUENCY. AND SYMBOLIZE. INSERT
C      TICKS.
C      C
C      C
C      C EXTERNAL REFERENCES
C      C
C      A4P      % ADJUSTED MSS COORD FOR PRINT/PLOT COORD
C      C
C      C GLOBAL DECLARATIONS
C      C
C      INCLUDE KOMOWH.LIST      % COMMON OUTPUT WINDOW PACKETS
C      INCLUDE KOMTBL.LIST      % COMMON FREQ/TICK TABLE
C      INCLUDE KOKKLS.LIST      % COMMON CLASSIFICATION INFO
C      INCLUDE PXBDEF.LIST      % DEFINE BUFFER STRUCTURE
C      INCLUDE KOSYH.LIST      % COMMON SYMBOL TABLE
C      INCLUDE WINDEF.LIST      % DEFINE WINDOW PACKETS
C      C
C      C LOCAL DECLARATIONS
C      C
C      PARAMETER NUMBFS=6      % NUMBER OF MPX BUFFERS
C      INTEGER MPXBUF(NWIXBF,NXBUFS)      % ARGUMENT
C      INTEGER IPRTF(1)      % ARGUMENT
C      INTEGER NBINSO(NUMBFS)      % BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER
C      INTEGER NBINCO      % BIN NUMBER OF COLUMN 0 FOR PRINT BUFFER
C      INTEGER MS100L,MS100H,MS100S      % MSA SAMPLE*100: LOW,HIGH,SPACING
C      INTEGER MSASAM      % MSA SAMPLE NUMBER
C      INTEGER IPBIN      % PRINT BUFFER BIN POINTER
C      INTEGER IPOTIC      % PRINT BUFFER BIN FOR TICK
C      REAL ADJSAM      % ADJUSTED SAMPLE NUMBER
C      INTEGER IPIXL1,IPIXL2,IPIXL3,
C      IPIXL4,IPIXL5,IPIXL6      % PIXEL VALUE FOR EACH BUFFER
C
C      C
C      C PROCEDURE
C      C
C      C
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICD13
009

```
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
  CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHIN))
  MS100L=ADJSAM*100.
  CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHAX))
  MS100H=ADJSAM*100.
  MS100S=MSA0MW(MSAM,WSP100)

C
C
C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C
  DO 150 NUMBUF=1,NXBUFS
    NBINS0(NUMBUF)=MPXBUF(PXLBIN,NUMBUF)-MPXBUF(PXLSAM,NUMBUF)
  150 CONTINUE

C
C
C SET PREAMBLE POINTERS AND BIN POINTER
C
  IPRTBF(PXRECN)=MPXBUF(PXRECN,1)
  IPRTBF(PXLINO)=IPLIN
  IPRTBF(PXCHAN)=0
  IPRTBF(PXQUAL)=0
  IPRTBF(PXBINT)='INT'
  IPRTBF(PXLBIN)=2
  IPRTBF(PXLCOL)=IPCHIN
  IPRTBF(PXHBIN)=IPCHAX-IPCHIN+2
  IPRTBF(PXHCOL)=IPCHAX
  IPRTBF(PXNOIN)=0
  IPRTBF(PXNODA)=0
  IPRTBF(PXLJO1)=0
  IPRTBF(PXHJO1)=0
  IPBIN=IPRTBF(PXLBIN)-1
  NBINCO=IPRTBF(PXLBIN)-IPRTBF(PXLCOL)  & FOR INSERTING TICKS

C
C
C RESAMPLE/SCREEN RADIANCE/COUNT FREQUENCY/LOOK UP SYMBOLS
C
  DO 400 MS100=MS100L,MS100H,MS100S
    MSASAM=MS100/100

C BUFFER 1
C
  IF ((MSASAM.LT.MPXBUF(PXLSAM,1)).OR.
& (MSASAM.GT.MPXBUF(PXHSA,1))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPXL1,
& MPXBUF(PXBINS,1),(MSASAM-NBINS0(1)))
  IF ((IPXL1.GE.MPXBUF(PXNODA,1)) GO TO 350  & NO DATA
& IF ((IPXL1.LT.LCVL01).OR.
& (IPXL1.GT.LCVH11)) GO TO 360  & OUT OF RAD LIMITS
& IF (NLINCH-I.EQ.0) GO TO 320

C
C BUFFER 2
C
  IF ((MSASAM.LT.MPXBUF(PXLSAM,2)).OR.
& (MSASAM.GT.MPXBUF(PXHSA,2))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPXL2,
```

```

      *
      *      MPXBUF(PXBINS,2).(MSASAM+NBINS0(2)))
      *      IF ((IPIXL2.LT.LCVLO(2)).OR.
      *      *      (IPIXL2.GT.LCVHI(2))) GO TO 360      * OUT OF RAD LIMITS
      *      *      IF (NLIMCH-2.EQ.0) GO TO 290
      *
      C
      C BUFFER 3
      C
      *      IF ((MSASAM.LT.MPXBUF(PXLSAM,3)).OR.
      *      *      (MSASAM.GT.MPXBUF(PXHSAM,3))) GO TO 350      * SAMPLE NOT IN BUFFER
      *      *      CALL GETBIN (IPIXL3.
      *      *      *      MPXBUF(PXBINS,3).(MSASAM+NBINS0(3)))
      *      *      IF ((IPIXL3.LT.LCVLO(3)).OR.
      *      *      *      (IPIXL3.GT.LCVHI(3))) GO TO 360      * OUT OF RAD LIMITS
      *      *      *      IF (NLIMCH-3.EQ.0) GO TO 260
      *
      C
      C BUFFER 4
      C
      *      IF ((MSASAM.LT.MPXBUF(PXLSAM,4)).OR.
      *      *      (MSASAM.GT.MPXBUF(PXHSAM,4))) GO TO 350      * SAMPLE NOT IN BUFFER
      *      *      CALL GETBIN (IPIXL4.
      *      *      *      MPXBUF(PXBINS,4).(MSASAM+NBINS0(4)))
      *      *      IF ((IPIXL4.LT.LCVLO(4)).OR.
      *      *      *      (IPIXL4.GT.LCVHI(4))) GO TO 360      * OUT OF RAD LIMITS
      *      *      *      IF (NLIMCH-4.EQ.0) GO TO 230
      *
      C
      C BUFFER 5
      C
      *      IF ((MSASAM.LT.MPXBUF(PXLSAM,5)).OR.
      *      *      (MSASAM.GT.MPXBUF(PXHSAM,5))) GO TO 350      * SAMPLE NOT IN BUFFER
      *      *      CALL GETBIN (IPIXL5.
      *      *      *      MPXBUF(PXBINS,5).(MSASAM+NBINS0(5)))
      *      *      IF ((IPIXL5.LT.LCVLO(5)).OR.
      *      *      *      (IPIXL5.GT.LCVHI(5))) GO TO 360      * OUT OF RAD LIMITS
      *      *      *      IF (NLIMCH-5.EQ.0) GO TO 200
      *
      C
      C BUFFER 6
      C
      *      IF ((MSASAM.LT.MPXBUF(PXLSAM,6)).OR.
      *      *      (MSASAM.GT.MPXBUF(PXHSAM,6))) GO TO 350      * SAMPLE NOT IN BUFFER
      *      *      CALL GETBIN (IPIXL6.
      *      *      *      MPXBUF(PXBINS,6).(MSASAM+NBINS0(6)))
      *      *      IF ((IPIXL6.LT.LCVLO(6)).OR.
      *      *      *      (IPIXL6.GT.LCVHI(6))) GO TO 360      * OUT OF RAD LIMITS
      *
      C
      C COUNT FREQUENCIES AND SYMBOLIZE
      C
      200      NFREQ(IPIXL6+1,6)=NFREQ(IPIXL6+1,6)+1
      230      NFREQ(IPIXL5+1,5)=NFREQ(IPIXL5+1,5)+1
      260      NFREQ(IPIXL4+1,4)=NFREQ(IPIXL4+1,4)+1
      290      NFREQ(IPIXL3+1,3)=NFREQ(IPIXL3+1,3)+1
      320      NFREQ(IPIXL2+1,2)=NFREQ(IPIXL2+1,2)+1
      *      NFREQ(IPIXL1+1,1)=NFREQ(IPIXL1+1,1)+1
      *      IPRTF(PXBINS+IPBIN)=KSYM(IPIXL1+1)
      *      GO TO 390
      350      IPRTF(PXBINS+IPBIN)=*      * NO DATA SYMBOL
      *      GO TO 390
  
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PIC013
011

```
380      IPRTBF(PXBINS+IPBIN)=' '      & OUT OF RADIANCE LIMITS
390      IPBIN=IPBIN+1
400 CONTINUE

C
C
C GET AND INSERT TICKS
C
500 IPBTIC=IPCTIC+NBINCO-1
   IF (IPLTIC.GT.IPLIN) GO TO 900      & SAVE TICK FOR SUBSEQUENT LINE
   IF (
     & (IPLTIC.LT.IPLIN).OR.           & TICK ABOVE WINDOW
     & (IPBTIC.LT.IPRTBF(PXLBIN)).OR.  & TICK LEFT OF WINDOW
     & (IPBTIC.GT.IPRTBF(PXHBIN)))     & TICK RIGHT OF WINDOW
     & GO TO 550                       & THEN IGNORE IT
   IF (JSYTIC.EQ.'*') GO TO 530      & ALWAYS INSERT PRIMARY TICKS
     IF (IPRTBF(PXBINS+IPBTIC).EQ.' ') GO TO 530
       IF (IPRTBF(PXBINS+IPBTIC).NE.'*') GO TO 550
         IF (IPRTBF(PXBINS+IPBTIC-1).EQ.'*')
           IPRTBF(PXBINS+IPBTIC-1)=' '      & HALO
         IF (IPRTBF(PXBINS+IPBTIC+1).EQ.'*')
           IPRTBF(PXBINS+IPBTIC+1)=' '      & HALO
530 IPRTBF(PXBINS+IPBTIC)=JSYTIC
550 CALL GETIC (IPLTIC,IPCTIC,JSYTIC)
     GO TO 500

C
C
900 RETURN
END
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICD14
002

```
CALL RDNOTE(
  = '(DISPLAY.',
  & CDSNCS(KLSTYP,(1),(3)),
  & ' NOT YET IMPLEMENTED)')
CALL NVIATO( PIC129.PICD19)
RETURN
END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE PICDIS  & DISPLAY CLASS (PHASE 9)  
      -----  
C  
C (E H SCHLOSSER)  
C  
C  
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED  
C -----  
C  
C      HDNOTE  
C      NVIATO  
C  
C  
C      EXTERNAL PIC129.PICDIS  
C      CALL TRACE  
C  
C  
C  
C      CALL HDNOTE('DISPLAY.CLASS NOT YET IMPLEMENTED')  
C      CALL NVIATO(PIC129.PICDIS)  
C      RETURN  
C      END
```


SUBROUTINE PIC019 8 DISPLAY MSS-DERIVED DATA (PHASE 9)

C
C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 10/05/75 ORIGINAL CODE
C E H SCHLOSSER LEC 08/27/78 UPGRADE DOCUMENTATION
C E H SCHLOSSER LEC 05/01/79 DON'T INCREMENT NMNDOM IF DIAGNOSTIC
C
C
C METHOD
C -----
C
C CHECK DIAGNOSTIC COUNTERS AND PREPARE FOR NEXT DISPLAY.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C NONE.
C
C
C EXTERNAL REFERENCES
C -----
C
C MDNOTE
C MDCLRW
C
C
C EXCEPTIONS
C -----
C
C NONE.
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
C INCLUDE NULCST.LIST 8 DEFINE NULL CHARACTER STRING
C EXTERNAL PIC000.NULSUB
C
C
C
C PROCEDURE
C -----
C
C CALL TRACE
C
C
C ON RETURN. CALL PIC000 TO GET COMMANDS
C
C CALL NVIATO(PIC000.NULSUB)
C
C

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICD19
002

C ANY DIAGNOSTICS???

C

980 IF(INDFATL.EQ.0) GO TO 920

CALL HDNOTE('FATAL ERRORS -- NO DISPLAY GENERATED')

GO TO 990

920 IF(INDHARN.EQ.0) GO TO 980

CALL HDNOTE('PREVIOUS WARNINGS -- NO DISPLAY GENERATED')

IF(MBATCH.EQ.0) WRITE(6,925)

925 FORMAT(4X, '...TRY AGAIN!')

CALL HDCLR(NULCST)

GO TO 980

C

C

C PREPARE FOR NEXT DISPLAY

C

990 NHNDOM=NHNDOW+1

C

C

C RETURN TO NEXT STATEMENT IN CALLING ROUTINE

C

990 RETURN

END

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DISHIS
001

```
      SUBROUTINE DISHIS( 3 HISTOGRAM PREVIOUSLY DISPLAYED DATA  
U (KOND)      3 1: FIRST 3 CHARS OF COMMAND 0: SPACES  
-----  
C  
C  
C (E W SCHLOSSER)  
C  
C  
      INCLUDE KONKQT.LIST  
      EXTERNAL PIC000.NULSUB  
C  
C  
      CALL NVIATO(PIC000.NULSUB)  
      CALL MDNOTE('HISTOGRAM COMMAND NOT YET IMPLEMENTED')  
      KOND=''  
      RETURN  
      END
```

SUBROUTINE PICEX1: 3 TERMINATION ROUTINE FOR PICTAB
U KOMD) 3 I: FIRST 3 CHARS OF COMMAND O: SPACES

```
C
C
C HISTORY
C -----
C
C   E H SCHLOSSER .EC   10/01/75   DESIGN/CODE/TEST
C   J C CRISP     LEC   12/31/79   UPDATE DOCUMENTATION
C   J C CRISP     LEMSCO 05/28/80   ONLY CALL IDERTS IF STRIP 4
C
C METHOD
C -----
C
C   CLOSE/VERIFY INPUT ERTS TAPE. ASK ABOUT PRINTER DISPLAYS. PRINT
C   DISPLAYS ONSITE. IF REQUESTED. ELSE DELETE ALTERNATE PRINT FILES.
C   TERMINATE PROGRAM.
C
C MACHINE-DEPENDENT CODE
C -----
C
C   NONE
C
C EXTERNAL REFERENCES
C -----
C
C   IDLU3      3 PRINT SHORT ID FOR LOGICAL UNIT 3
C   DLETPR    3 DELETE ALTERNATE PRINT FILES
C   CLOSE3    3 CLOSE AND VERIFY 3EOF ON INPUT ERTS TAPE
C   IDERTS    3 PRINT COMPLETE ERTS SCENE IDENTIFICATION
C   READS     3 FILL BUFFER FOR UNIT 5
C   CLOSPR    3 CLOSE ALTERNATE PRINT FILES
C   PSTOP     3 PROGRAM TERMINATION
C
C EXCEPTIONS
C -----
C
C   NONE
C
C GLOBAL DECLARATIONS
C -----
C
C   INCLUDE KOMXQT.LIST 3 COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
C   INCLUDE KOMNER.LIST 3 COMMON ERTS SCENE PARAMETERS
C   INCLUDE NULCST.LIST 3 DEFINE NULL CHARACTER STRING
C
C LOCAL DECLARATIONS
C -----
C
```

```
          INTEGER NORY      & RESPONSE TO PRINT FILES ONSITE
C
C
C PROCEDURE
C -----
C
          CALL TRACE
C
C
C GET OPTIONAL 'QUICK' SPECIFICATION
C
          IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE
          ITEMP=0
          CALL GETSKH(ITEMP,(3), NULCST)
          IF(ITEMP.NE.'QUI') GO TO 120
          WRITE(6,105)
          105 FORMAT(4X,'**QUICK PROGRAM TERMINATION')
          GO TO 150
C
C
C CLOSE/VERIFY INPUT ERTS TAPE
C
          120 WRITE(6,125)
          125 FORMAT(4X,'**PROGRAM TERMINATION')
          CALL CLOSE3
C
C
C IDENTIFY ERTS SCENE WITH PITCH & ROLL IF JUST EXTRACTED FROM SIAT
C
          IF((NCCT.NE.4).OR.(ABS(PITDEG).GT.9.)) GO TO 150
          WRITE(6,145)
          145   FORMAT(1X)      & SKIP LINE
          CALL IDLU3( 6)
          CALL IDERTS( 6)
          WRITE(10,145)
          CALL IDLU3( 10)
          CALL IDERTS( 10)
C
C
C BATCH RUN WITH FATAL ERRORS OR ANY DEMAND RUN -- ASK ABOUT PRINTER DISPLAY(S)
C
          150 IF((MBATCH.EQ.1).AND.(NOFATL.EQ.0)) GO TO 200      & BATCH & OK
          IF(NOFATL.NE.0) CALL MOWARN(
          - 'FATAL ERROR(S) ENCOUNTERED -- DISPLAY(S) ARE DEFECTIVE')
          WRITE(6,165)
          165 FORMAT(' OUTPUT DISPLAY(S) ON LINE PRINTER?')
          IF(KOMD.NE.'EOF5') CALL READ5(L5STAT, ' ')
          NORY='N'
          CALL GETSKH(NORY,(1), NULCST)
          IF(NORY.NE.'Y') GO TO 700
C
C
C PRINT DISPLAY(S) ONSITE
C
          200 NHNDOW=MAX0(0,NHNDOW-1)
          WRITE(6,245) NHNDOW
```

ORIGINAL PAGE IS
OF POOR QUALITY

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICEX1
003

245 FORMAT(IX.14,' DISPLAY(S) PRINTED')
CALL CLOSPR
00 TO 900

C
C
C DON'T PRINT DISPLAY(S) ONSITE
C
C 700 CALL DLETPR
C
C
C TERMINATE PROGRAM
C
C 900 CALL PSTOP('***PLEASE 8FREE 3. OR 8REWIND 3. OR 8LOCATE 3.')

C
C
C PSTOP DOES NOT RETURN
C
END


```
INCLUDE KOMFIT.LIST      & COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE WINDEF.LIST      & DEFINE WINDOW PACKETS
INCLUDE KOMINW.LIST      & DEFINE INPUT WINDOW PACKET
INCLUDE KOMOHV.LIST      & DEFINE OUTPUT WINDOW PACKET
```

```
C
C
C LOCAL DECLARATIONS
C -----
C
C     NONE
C
C
C PROCEDURE
C -----
C
C     CALL TRACE
C
C
C CHECK COMMAND VALIDITY
C
C     IF(NWINDOW.EQ.0) CALL MDHARN('INVALID DEFAULT COMMAND')
C
C
C DRAIN SPECS FOR CURRENT COMMAND
C
C     KLSTYP='RAD'      & DEFAULT IS FACTORING OF RADIANCE
C     CALL GETSIN(ITEMP,+1,-1,'EXTRA FACTOR SPECIFICATION --')
C
C
C CHECK FOR SPACING LESS THAN 1 -- (SPACING SCALED BY 100)
C
C     IF ((MSAOWH(WLIN,WSP100).GE.100).AND.
C &      (MSAOWH(WSAM,WSP100).GE.100)) GO TO 200
C     CALL MDHARN ('REQUESTED SPACING LESS THAN 1.')
C     GO TO 800
C
C
C CALIBRATE SPACING/CHANNELS/WINDOW
C
C 200 CALL CALSPA
C     CALL CALCHA
C     CALL CALWIN(0.)
C
C
C CROP OUTPUT WINDOW TO FIT INPUT WINDOW
C
C     CALL CROPOW(2500,3500)
C
C
C CHECK FOR DIAGNOSTICS
C
C     IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE
C     IF(NDOTL.NE.0) GO TO 800
C
C
C
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICFAC
003

```
C CLEAR WINDOW NUMBER
C
  IF(NHNDOW.LT.0) CALL OPRPIC      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
  NHNDOW=1ABS(NHNDOW)
C
C
C PRINT WINDOW HEADING FOR UNIT 8
C
  IF(MBATCH.NE.0) CALL HOUNIT(4.8)
  WRITE(8.415) NHNDOW.MTERAL
  *15 FORMAT(' WINDOW NUMBER ',J3.0X,'FACTOR',6X.4A6)
  CALL IDLU3(8)
  CALL IDERT(8)
  CALL IDCPIC(8)
C
C
C PRINT WINDOW HEADING FOR UNIT 10
C
  *50 CALL HOUNIT(4.10)
  CALL IDLU3(10)
  CALL IDERT(10)
  CALL IDCPIC(10)
C
C
C NAME SUBROUTINE TO DO FACTORING
C
  CALL NVIATO( PIC345,PICFA3)      & NEXT CALL IS TO PICFA3
  GO TO 900
C
C
C WARNINGS ENCOUNTERED -- NEXT CALL IS TO PICFA9
C
  800 CALL NVIATO( PIC129,PICFA9)
  KLSTYP=0      & NO FACTORING PERFORMED
C
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
  900 KOND=' '
  RETURN
  END
```



```

C
  INCLUDE KOMXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
  INCLUDE KOMLU3.LIST     & PACKET POINTERS FOR UNIT 3
  INCLUDE KOMTBL.LIST     & COMMON FACTOR TABLE
  INCLUDE KOMKLS.LIST     & COMMON CLASSIFICATION INFO
  DOUBLE PRECISION CBS%IN  & VARIABLE-LENGTH CHAR STRING FOR INTEGER
C
C LOCAL DECLARATIONS
C -----
C
  PARAMETER NRCDIM=7
  REAL TEST(10,3)/
  & 7..4..8..8..8..7..5..9..7..8..
  & 4..1..3..8..5..2..3..5..4..2..
  & 3..8..5..1..7..9..3..8..5..2./  & TEST DATA FROM COOLEY & LOHNES
C                                     & (1971),P 110
  DOUBLE PRECISION DSUM(NRCDIM)      & SUMS
  DOUBLE PRECISION DSPROD(NRCDIM,NRCDIM) & SUMS OF PRODUCTS
  REAL EVALRR(NRCDIM,NRCDIM)        & EIGENVALUES***-0.5 (ON DIAGONAL)
  INTEGER NROW,NCOL                 & ROW/COLUMN NUMBER
C
C PROCEDURE
C -----
C
  CALL TRACE
C
C FLAG KTABLE AS DESTROYED
C
  KTBLTY=0
C
C INITIALIZE NUMBER OF CHANNELS AND FACTORS FOR MSS DATA
C
  IF (LUSSEQ(1).EQ.'NUL') GO TO 180  & NO TAPE--USE TEST DATA
  IF (NLIMCH.LT.2) CALL MDWARN ('ONLY ONE CHANNEL REQUESTED')
  KCHACO=NLIMCH
  KFACCO=KCHACO
  IF (INDTOTL.NE.0) GO TO 900
C
C READ SCAN LINES--COMPUTE SUMS AND SUMS OF PRODUCTS
C
  CALL DSSPR3 (DSUM,DSPROD,(NRCDIM),KPIXCO, KCHACO)
  IF (KPIXCO.GE.100) GO TO 200
  CALL MDWARN ('PIXEL COUNT OF '.CBS%INIKPIXCO.3).' TOO SMALL')
  GO TO 900
C
C COMPUTE NUMBER OF PIXELS, CHANNELS, FACTORS FOR TEST DATA
C
  180 KPIXCO=10
  KCHACO=3
  KFACCO=3

```

```
C
C
C COMPUTE SUMS & SUMS OF PRODUCTS FOR TEST DATA
C   CALL DSSPR(DSUM,DSPROD,(NRCDIM), TEST,KPIXCO,KCHACO)
C
C COMPUTE CORRELATIONS, MEANS, STANDARD DEVIATIONS
C
200 CALL DCORLT(CORREL,CHMEAN,CHSTD,
-           KPIXCO,DSUM,DSPROD,KCHACO,NRCDIM)
  DO 240 NSTD=1,KCHACO
    IF (CHSTD(NSTD).GE.1.) GO TO 240
    CALL HDWARN ('STD DEVIATION LESS THAN 1. FOR CHAN ',
&             CBS4IN(NSTD,2))
    GO TO 900
240 CONTINUE
C
C
C DUPLICATE CORRELATION MATRIX
C
  DO 260 NROW=1,KCHACO
    DO 260 NCOL=1,KCHACO
      EIOVAL(NROW,NCOL)=CORREL(NROW,NCOL)
260 CONTINUE
C
C
C COMPUTE EIGENVALUES & EIGENVECTORS & SORT IN ORDER OF DESCENDING EIGENVALUES
C
  IT=50
  CALL JACMX(EIOVAL,EIOVEC,NRCDIM,KCHACO,5.E-7,IT,$800,1)
  CALL EISRTD(EIOVAL,EIOVEC,KCHACO,NRCDIM)
C
C
C COMPUTE PERCENT OF VARIANCE ACCOUNTED FOR BY EACH FACTOR
C
  CALL DOPCNT(EIOVAL,KCHACO,KFACCO,NRCDIM,NRCDIM,PCTVAR)
C
C
C COMPUTE FACTOR STRUCTURE (FSTRUC) = (EIOVAL**-.5) (EIOVEC)
C (FSTRUC) IS CORRELATION MATRIX BETWEEN CHANNELS (ROWS) & FACTORS (COLUMNS)
C
  CALL DOSQRT(EIOVAL,KCHACO,KFACCO,NRCDIM,NRCDIM,EVALRR)
  CALL MXMLT(EIOVEC,EVALRR,FSTRUC,KCHACO,KCHACO,KFACCO,
&           NRCDIM,NRCDIM)
C
C
C COMPUTE NORMALIZED FACTOR COEFFICIENTS (FCNORM)=(EIOVAL**-.5)(EIOVEC)
C
  CALL DORECP(EVALRR,KCHACO,KFACCO,NRCDIM,NRCDIM,EVALRR) & RECIPROCAL
  CALL MXMLT(EIOVEC,EVALRR,FCNORM,KCHACO,KCHACO,KFACCO,
&           NRCDIM,NRCDIM)
C
C
C MARK KTABLE AS CONTAINING FACTOR PARAMETERS FROM PRINCIPAL FACTOR ANALYSIS
C
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICFAS
004

KTBLTY='FACT'
KTBLNW=NMNDOM
00 TO 900

C
C
C FLAG ERROR DETECTED BY EIGENVALUE/EIGENVECTOR ROUTINE
C
C 000 CALL MDHARN('MAX ITERATIONS IN EIGEN')
C
C
C 900 CALL NVIATO(PIC129.PICFAS)
C
C
C RETURN
C END

```
      SUBROUTINE PICFAS 8 FACTOR RAN HSS CHANNELS (PHASE 8)  
-----  
C  
C  
C  
C      M L BROWN      LEC  01/23/78      ALGORITHM CODING  
C      J C CRISP      LEC  12/08/78      UPGRADE DOCUMENTATION  
C  
C  
C METHOD  
C -----  
C  
C      PRINT RESULTS OF FACTORING FROM KTABLE.  
C  
C  
C MACHINE-DEPENDENT CODE  
C -----  
C  
C      NONE  
C  
C  
C EXTERNAL REFERENCES  
C -----  
C  
C      MONOTE  
C      NVIATO  
C      MATPRT  
C      FACPRT  
C      EXTERNAL PIC000.NULSUB  
C  
C  
C GLOBAL DECLARATIONS  
C -----  
C  
C      INCLUDE KOMXQT.LIST  
C      INCLUDE KOMTBL.LIST  
C  
C  
C PROCEDURE  
C -----  
C  
C      CALL TRACE  
C  
C      CALL NVIATO(PIC000.NULSUB)  
C  
C  
C CHECK FOR DIAGNOSTICS  
C  
C      IF(KTBLTY.NE.'FACT') GO TO 800  
C      IF(NDOTL.NE.0) GO TO 800  
C  
C  
C      125 FORMAT(' NUMBER OF PIXELS'/X,112/)  
C      WRITE(6,125) KPIXCO  
C  
C  
C      125 FORMAT(' CHANNEL STANDARD DEVIATION2')
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICFAD
002

```
      WRITE(6,145)
      CALL MATPRT(6,CHSTD,' .1,KFACCO,1.7)
C
C
225 FORMAT(' CHANNEL MEANS')
      WRITE(6,225)
      CALL MATPRT(6,CHMEAN,' .1,KFACCO,1.7)
C
C
245 FORMAT(' CORRELATION BETWEEN CHANNELS')
      WRITE(6,245)
      CALL MATPRT(6,CORREL,' CHAN',KCHACO,KFACCO,7.7)
C
C
325 FORMAT(' EIGENVALUES (DIAGONAL)')
      WRITE(6,325)
      CALL MATPRT(6,EIOVAL,' ROW',KCHACO,KFACCO,7.7)
C
C
345 FORMAT(' EIGENVECTORS')
      WRITE(6,345)
      CALL MATPRT(6,EIOVEC,' ROW',KCHACO,KFACCO,7.7)
C
C
365 FORMAT(' PRINCIPAL FACTOR ANALYSIS (UNROTATED) ---')
      WRITE(6,365)
C
C
385 FORMAT(' INDIVIDUAL (1) & CUMULATIVE (2) % OF VARIANCE BY FACTOR')
      WRITE(6,385)
      CALL MATPRT(6,PCTVAR,' %',2,KFACCO,2.7)
C
C
      CALL FACPRN(FSTRUC,FCNORM,FCORIG,CHSTD,CHMEAN,KCHACO,KFACCO,7.7)
      GO TO 890
C
C
C CHECK DIAGNOSTIC COUNTERS
C
800 IF(NDWARN.EQ.0) GO TO 820
      CALL MDNOTE('PREVIOUS WARNINGS -- NO FACTORING PERFORMED')
      IF(MBATCH.EQ.0) WRITE(6,815)
      815 FORMAT(' ...TRY AGAIN')
      GO TO 900
820 IF(NDFATL.EQ.0) GO TO 850
      CALL MDNOTE('PREVIOUS FATAL ERRORS -- NO FACTORING PERFORMED')
      GO TO 900
850 IF(MCHECK.EQ.0) GO TO 890
      CALL MDNOTE('CHECKOUT MODE -- NO FACTORING PERFORMED')
      GO TO 900
C
C
C PREPARE FOR NEXT DISPLAY
C
890 NWINDOW=NWINDOW+1
C
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICFAS
003

C
C CLEAR WARNINGS & RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
000 CALL MDCLRW(-0)
RETURN
END

C	CONDITION	ACTION	DIAGNOSTIC
C	-----	-----	-----
C	PROCESSING DEFAULT COMMANDS		
C	(NWINDOW=0)	NONE	WARNING
C	KLSTYP IN COMMON UNDEFINED	KLSTYP='RAD'	NONE
C	KLSTYP SPECIFICATION MISSING	USE COMMON KLSTYP	NONE
C	KLSTYP SPECIFICATION INVALID	NONE	WARNING
C	EXTRA SPECIFICATION	NONE	WARNING
C	LIMIT CHANNEL VALUE RANGE IS NULL		
C	(LCVLO1>LCVHI1)	NONE	WARNING
C	DATA/CHECKOUT MODE	'TO' ROUTINE IS NULSUB	NONE
C	WARNING(S) OR FATAL ERROR(S)	'TO' ROUTINE IS PICL19	NONE
C			
C			
C	GLOBAL DECLARATIONS		
C	-----		
C			
	INCLUDE KOXQT.LIST	% COMMON PROGRAM EXECUTION SWITCHES. COUNTERS	
	INCLUDE KOXKLS.LIST	% COMMON CLASSIFICATION SPECTRAL LIMITS	
	INCLUDE KOMTBL.LIST	% COMMON TICK/FREQ/FACTOR TABLE	
	INCLUDE WINDEF.LIST	% DEFINE STRUCTURE OF WINDOW PACKETS	
	INCLUDE KOHOWH.LIST	% COMMON OUTPUT WINDOW PACKETS	
	INCLUDE NULCST.LIST	% DEFINE NULL CHARACTER STRING	
C			
C			
C	LOCAL DECLARATIONS		
C	-----		
C			
	INTEGER KLSAVE	% SAVE LOCATION FOR KLSTYP	
	INTEGER INTMP	% TEMPORARY	
	INTEGER NPRLIN	% NUMBER OF PRINT LINES IN LIST	
	INTEGER NPRCOL	% NUMBER OF LOGICAL PPD COLUMNS IN LIST	
	INTEGER LUNALT	% LOGICAL UNIT NUMBER TO SPOOL PRINTER OUTPUT	
	INTEGER NPROCIF	% NUMBER OF LOGICAL PPD COLUMNS IN ONE FILE	
	INTEGER NODMAX	% MAXIMUM # OF NODES IN PPDOWH	
	INTEGER NOD	% # OF NODE IN PPDOWH/PPDOWH	
	INTEGER PPDOWH(2,HHO)	% TEMPORARY PPDOWH	
C			
C			
C	PROCEDURE		
C	-----		
C			
	CALL TRACE		
C			
C			
C	GET LIST TYPE		
C			
	KTBLTY='NUL'	% MARK OLD FREQ TABLE AS DESTROYED	
	IF(NWINDOW.EQ.0) CALL HDWARN('INVALID DEFAULT COMMAND')		
	KLSAVE=KLSTYP	% SAVE PREVIOUS LIST TYPE	
	IF((KLSTYP.NE.'GRA').AND.		
	& (KLSTYP.NE.'LAP').AND.		
	& (KLSTYP.NE.'VAR').AND.		
	& (KLSTYP.NE.'CLA'))	% IF UNDEFINED ...	
	& KLSTYP='RAD'	% ... THEN MAKE IT RADIANCE) ...	

```
CALL GETSKH(KLSTYP.(3), NULCST) & ... UNLESS SPECIFIED BY LIS(?) CMD
C
C CHECK LIST TYPE
C
IF(KLSTYP.NE.'RAD') GO TO 240          & RADIANCE?
CALL NVIATO( PIC345.PICL13) & NEXT CALL IS TO PICL13
GO TO 300
240 IF((KLSTYP.NE.'GRA').AND.          & GRAIDENT?
& (KLSTYP.NE.'LAP').AND.             & LAPILACIAN?
& (KLSTYP.NE.'VAR')) GO TO 250      & VARIANCE?
CALL NVIATO( PIC345.PICL14) & NEXT CALL IS TO PICL14
GO TO 300
250 IF(KLSTYP.NE.'CLA') GO TO 280     & CLAISS?
CALL NVIATO( PIC345.PICL15) & NEXT CALL IS TO PICL15
GO TO 300
280 CALL WARN5( 'BAD LIST TYPE --')
KLSTYP=KLSAVE & RESTORE PREVIOUS LIST TYPE
C
C DRAIN SPECS FOR CURRENT COMMAND
C
300 CALL GETSIN(INTEMP, +1.-1.'EXTRA LIST SPECIFICATION --')
C
C CHECK RADIANCE LIMITS
C
IF(LCVLO1.GT.LCVHI1) CALL MDHARN: 'NO RADIANCE LIMITS'
IF(MDATAC.NE.0) GO TO 900 & DATA/CHECKOUT MODE
C
C CALIBRATE SPACING/CHANNELS
C
CALL CALSP/
CALL CALCHA
C
C SAVE THE PHYSICAL PPD VERTEX COLUMNS
C
NODMAX=800L (PPDOWW(WUSED,WHEAD))
DO 350 NOD=WVER.NODMAX
PPDOW(WCOL,NOD)=PPDOW(WCOL,NOD)
350 CONTINUE
C
C TRANSFORM TO LOGICAL PPD VERTEX COLUMNS FROM PHYSICAL
C
DO 375 NOD=WVER.NODMAX
PPDOW(WCOL,NOD)=PPDOW(WCOL,NOD)/3
375 CONTINUE
C
C CALIBRATE OUTPUT WINDOW ENVELOPE USING LOGICAL PPD VERTEX COLUMNS
C
CALL CALWIN( 0.)
C
```

```
C
C CROP OUTPUT WINDOW TO FIT INPUT WINDOW & ALT PRINT FILE(S)
C   (3 PHYSICAL PPD COLUMNS PER LOGICAL PPD COLUMN)
C
C   NPRCIF=(KPAGE-4-2)/3      & NUMBER OF LOGICAL PPD COLUMNS PER FILE
C                               & EXCLUDING LEFT MARGIN & 2 NEAT LINES
C   CALL CROPM( 1000,MALTM*NPRCIF)
C
C .STORE THE PHYSICAL PPD VERTEX COLUMNS
C
C   DO 400 NOD=MVER,NODMAX
C     PPDOWH(WCOL,NOD)=PPDTOW(WCOL,NOD)
C 400 CONTINUE
C
C
C C CHECK FOR DIAGNOSTICS
C
C   IF (NDTOTL.NE.0) GO TO 900
C
C
C C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
C   IF(NHNDOW.LT.0) CALL OPRPIC      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
C   NHNDOW=ABS(NHNDOW)
C   NPAGE=0
C
C
C C COMPUTE LOGICAL SIZE OF PRINT WINDOW FROM LOGICAL PPD ENVELOPE
C
C   NPRLIN=FIX(PPDOWH(WLIN,WMAX))-FIX(PPDOWH(WLIN,WMIN))+1
C   NPRCOL=FIX(PPDOWH(WCOL,WMAX))-FIX(PPDOWH(WCOL,WMIN))+1
C
C
C C CHECK FOR DIAGNOSTICS
C
C   IF(NDTOTL.NE.0) GO TO 900
C
C
C C PRINT WINDOW HEADING FOR UNIT 6
C
C   WRITE(6,415) NHNDOW,MTERAL
C 415 FORMAT(' WINDOW NUMBER ',J3.6X,'LIST',6X,4A6)
C   CALL IDLU3( 6)
C   CALL IDERT( 6)
C   CALL IDCPIC( 6)
C
C
C C PRINT WINDOW HEADING FOR EACH ALTERNATE PRINT FILE IN LOGICAL PPD ENVELOPE
C
C   LUNALT=10
C   DO 460 N=1,NPRCOL,NPRCIF
C     CALL HUNIT( 4,LUNALT)
C     WRITE(LUNALT,415) NHNDOW,MTERAL
C     CALL IDLU3( LUNALT)
C     CALL IDERT( LUNALT)
```

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICL15
005

```
          CALL IDCPIC( LUNALT)
          LUNALT=LUNALT+1
480 CONTINUE
C
C
C ANY DIAGNOSTICS???
C
900 IF(NDTOTL.EQ.0) GO TO 990
      IF(MDATAC.NE.0) CALL NVIATOC( PIC000.NULSUB)      8 DATA/CHECKOUT
      IF(MDATAC.EQ.0) CALL NVIATOC( PIC129.PICL19)
C
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KOMD=1
      RETURN
      END
```


DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

PICL13
 002

C	'BADR'	'?'	NO DATA (':::')	NONE	PRINT LINE
C	'BADF'	N/A	N/A	FATAL	RETURN
C	'OFL'	N/A	N/A	FATAL	RETURN

C GLOBAL DECLARATIONS

C -----

C
 C INCLUDE KOMXQT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
 C INCLUDE KOMKLS.LIST & COMMON CLASSIFICATION INFO
 C INCLUDE WINDEF.LIST & DEFINE STRUCTURE OF WINDOW PACKETS
 C INCLUDE KOMOHM.LIST & COMMON OUTPUT WINDOW PACKETS
 C INCLUDE KOMTBL.LIST & COMMON TICK/FREQ TABLE
 C INCLUDE PXBDEF.LIST & DEFINE BUFFER STRUCTURE
 C INCLUDE PICDEF.LIST & DEFINE PICTAB PARAMETERS
 C INCLUDE MAXINT.LIST & MAXIMUM INTEGER VALUE

C
 C LOCAL DECLARATIONS

C -----

C
 C PARAMETER NXBUFS=6 & # OF MSS PIXEL BUFFERS IN ARRAY
 C
 C INTEGERS IN MSA BUF = #INTS PREAMBLE + (#BINS*3)/4 + (#EXTRA BYTES*3)/4
 C PARAMETER NWIXBF = (PXBINS-1) + (35*8*3)/4 + (19*3)/4
 C
 C WDS PRT BUF=WDS PREAM+(MAX FILES*(MAX COLS/PO-NEATLN COLS-MARG COLS))*4
 C PARAMETER NWIPBF=(PXBINS-1)*(MALTHI + (KPAQHI- 2 - 3)) + 4
 C
 C PARAMETER NFRQCH=5 & # WDS PER CHANNEL IN LOCAL FREQ TABLE
 C PARAMETER NFRQSZ=128 & # CHANNELS IN LOCAL FREQ TABLE
 C INTEGER MPXBUF(NWIXBF,NXBUFS) & ARRAY OF MSS PIXEL BUFFERS
 C INTEGER IPRTBF(NWIPBF) & PRINT BUFFER
 C INTEGER NFREQ(NFRQSZ,NFRQCH) & LOCAL FREQUENCY TABLE (SCOPE INCLUDES
 C & INTERNAL ROUTINE RESSYM)
 C
 C INTEGER IPLIN & PRINT LINE
 C INTEGER IPCHIN,IPCHMAX & MINIMUM, MAXIMUM PRINT COLUMN
 C REAL ADJLIN,ADJSAM & ADJUSTED LINE AND SAMPLE
 C INTEGER ML100L,ML100H,ML100S & MSA LINE*100: LOW,HIGH,SPACING
 C INTEGER MSALIN & MSA LINE NUMBER
 C INTEGER ISTAT & I/O STATUS
 C INTEGER NTLCHR & NEAT LINE CHARACTER
 C INTEGER MRQLCS,MRQRCS & LEFT & RIGHT MARGIN CHAR STRING
 C INTEGER IPLMIN,IPLMAX & MINIMUM AND MAXIMUM PRINT LINE
 C INTEGER ML100 & MSA LINE*100
 C INTEGER MSASLO,MSASHI & LOW & HIGH MSA SAMPLE NUMBER
 C INTEGER NPRLIN,NPRCOL & # PRINT LINES, COLUMNS
 C INTEGER LASTLN & LAST SCAN LINE READ
 C INTEGER I,K & DO LOOP INDEXES

C
 C PROCEDURE

C -----

C
 C CALL TRACE

```
C
C INITIALIZE MINIMUM AND MAXIMUM PRINT LINES AND COLUMNS
C
      IPLMIN=PPDOWNH(WLIN.WMIN)
      IPLMAX=PPDOWNH(WLIN.WMAX)
      IPCMIN=PPDOWNH(WCOL.WMIN)
      IPCMAX=PPDOWNH(WCOL.WMAX)
C
C
C CALCULATE THE NUMBER OF PRINT LINES, COLUMNS TO BE PRINTED
C
      NPRLIN=IPLMAX-IPLMIN+1
      NPRCOL=(IPCMAX-IPCMIN+1)*3
C
C
C CLEAR LOCAL FREQUENCY TABLE
C
      DO 250 I=1.NLIMCH
        DO 200 K=1.NFROSZ
          NFREQ(K,I)=0
      200 CONTINUE
      250 CONTINUE
C
C
C INITIALIZE LOW AND HIGH LINES AND SPACING
C
      CALL A4P (ADJLIN,ADJSAM, FLOAT(IPLMIN),1.)
      ML100L=ADJLIN*100.
      CALL A4P (ADJLIN,ADJSAM, FLOAT(IPLMAX),1.)
      ML100H=ADJLIN*100.
      ML100S=MSAOWH(WLIN.WSP100)
C
C
C POSITION AT TOP OF WINDOW
C
      MSALIN=ML100L/100
      CALL GETRAD (MPXBUF,(0),(NXBUFS),ISTAT, MSALIN,0.0)
      IF (ISTAT.NE.'BADF') GO TO 300
      CALL MDFATL ('BADF (BAD FILE) ON UNIT 3')
      CALL ERPRTA (
B      '10          '.2.NI4NC(28)'.*1/O ERROR -- IGNORE OUTPUT*'
      GO TO 900
C
C
C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE BEFORE 1ST PRINT LINE
C
      300 CALL SAMSCL (IPRTBF,(IPLMIN-1),(IPCMIN),(IPCMAX))
      IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
B      CALL PROVFI (8.' .4.' .4.' .0.' .'.*1111'*.IPRTBF)
      CALL PROVFI (10.' .4.' .4.' .'.*11111'*.IPRTBF)
C
C
C READ, MASK, RESAMPLE, SCREEN, COUNT FREQUENCY AND PRINT SCAN LINES
C
      IPLIN=IPLMIN
      LASTLN=-MAXINT
```



```

DO 350 ML100=ML100L,ML100H,ML100S
  NTLCHR=';'
  MSALIN=ML100/100
  CALL A4P (ADJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCHIN))
  MSASLO=ADJSAM
  CALL A4P (ADJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCMAX))
  MSASHI=ADJSAM
  IF (MSALIN.NE.LASTLN) CALL GETRAD (MPXBUF,(NHIXBF),(NXBUFS),
    & ISTAT, MSALIN,MSASLO,MSASHI)
  LASTLN=MSALIN
  IF ((ISTAT.NE.'BADF').AND.(ISTAT.NE.'OFL')) GO TO 320
  CALL MDFATL (' CBS4CS(ISTAT,1,4),
    & ' WHILE READING ON UNIT 3')
  CALL ERPRTA ('10 '2.6.
    & '1/O ERROR - IGNORE OUTPUT')
  GO TO 900
320 IF (ISTAT.EQ.'BADR') NTLCHR='?'
  CALL MSKPIX (MPXBUF(1,1), MPXBUF(1,1))
  IF ((MPXBUF(PXBINT,1).EQ.'BYT').OR.
    & (MPXBUF(PXBINT,1).EQ.'CHR').OR.
    & (MPXBUF(PXBINT,1).EQ.'INT').OR.
    & (MPXBUF(PXBINT,1).EQ.'NUL')) GO TO 330
  CALL MDFATL('INVALID BIN TYPE ',
    & CBS4CS(MPXBUF(PXBINT,1),1,3),' IN PICL13')
  GO TO 900
330 IF (MPXBUF(PXBINT,1).EQ.'BYT') CALL RESCRN (IPRTBF,(IPLIN),
    & (IPCHIN),(IPCMAX), MPXBUF,(NHIXBF),(NXBUFS).GETBYT)
  IF (MPXBUF(PXBINT,1).EQ.'CHR') CALL RESCRN (IPRTBF,(IPLIN),
    & (IPCHIN),(IPCMAX), MPXBUF,(NHIXBF),(NXBUFS).GETICE)
  IF (MPXBUF(PXBINT,1).EQ.'INT') CALL RESCRN (IPRTBF,(IPLIN),
    & (IPCHIN),(IPCMAX), MPXBUF,(NHIXBF),(NXBUFS).GETINT)
  IF (MPXBUF(PXBINT,1).EQ.'NUL') CALL RESCRN (IPRTBF,(IPLIN),
    & (IPCHIN),(IPCMAX), MPXBUF,(NHIXBF),(NXBUFS).GETNUL)
  CALL CST4IN (MRGLCS,(1),(6), MSALIN,4,'0')
  MRGRCS=MRGLCS
  IF ((MBATCH.EQ.3).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
    & CALL PROVFI (6,MRGLCS,4,'0,NTLCHR,'1*****',IPRTBF)
  CALL PROVFI (10,'4','4','11111',IPRTBF)
  IPLIN=IPLIN+1
350 CONTINUE
C
C
C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE AFTER LAST PRINT LINE
C
  CALL SAMSCL (IPRTBF,(IPLMAX+1),(IPCHIN),(IPCMAX))
  IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
    & CALL PROVFI (6,'4','0','1*****',IPRTBF)
  CALL PROVFI (10,'4','4','11111',IPRTBF)
  CALL PROVFI (10,'4','4','31111',IPRTBF)
C
C
C MOVE DATA FROM LOCAL FREQ TABLE TO COMMON TABLE
C
  KTBLTY='FREQ'
  KTBLNW=NHNDOW
  DO 600 I=1,NLIMCH

```

```
      DO 550 K=1,NFREQ2
      KFREQ(K,1)=NFREQ(K,1)
550   CONTINUE
600   CONTINUE
C
C   NEXT CALL IS TO PICL19
C
C   900 CALL NVIATO ( PIC129,PICL19)
      RETURN
C
C
C
C
C
C
C   INTERNAL
C   SUBROUTINE SAHSC(  ) GENERATE SAMPLE SCALE AND BORDER
C   O IPRBUF.  ) PRINT BUFFER
C   ( IPLIN.  ) PRINT LINE
C   ( IPCHIN. ) MINIMUM PRINT COLUMN (LOGICAL)
C   ( IPCMAX) ) MAXIMUM PRINT COLUMN (LOGICAL)
C
C
C   HISTORY
C
C   D A BECK      LEC      11/07/79      REQUIREMENTS
C   D A BECK      LEC      11/07/79      ALGORITHM DESIGN
C   D A BECK      LEC      11/07/79      ALGORITHM CODING
C
C
C   METHOD
C
C   INITIALIZE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER
C   PREAMBLE. ENCODE SAMPLE NUMBERS AND PUT COLON, STRING, AND
C   COLON IN PRINT BUFFER.
C   NOTE, 3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN.
C
C
C   MACHINE-DEPENDENT CODE
C
C   ASSUMES 6 CHARS PER INTEGER BIN
C
C
C   EXTERNAL REFERENCES
C
C   ANP          ) ADJUSTED COORD FOR PRINT/PLOT COORD
C   PUTCHR       ) PUT CHAR IN CHAR STRING
C   CST4IN       ) CHARACTER STRING FOR INTEGER
C
C
C   EXCEPTIONS
C
C   ASSUMES 6 CHARACTERS PER INTEGER BIN.
C
```

```

C
C GLOBAL DECLARATIONS
C
C   INCLUDE KOMOHV.LIST      & COMMON OUTPUT WINDOW PACKETS
C   INCLUDE MINDEF.LIST     & DEFINE STRUCTURE OF WINDOW PACKETS
C   INCLUDE PXDEF.LIST      & DEFINE PIXEL BUFFER STRUCTURE
C
C LOCAL DECLARATIONS
C
C   INTEGER IPRBF(1)        & ARGUMENT
C   REAL ADJLIN,ADJSAM      & ADJUSTED LINE. SAMPLE
C   INTEGER IPBIN           & POINTER TO PRINT BIN
C   INTEGER MSASAM         & SAMPLE NUMBER
C   INTEGER MS100L,MS100H,MS100S & MSA SAMPLE*100: LOW,HIGH,SPACING
C
C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
C   CALL A4P(ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHIN))
C   MS100L=ADJSAM*100.
C   CALL A4P(ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHAX))
C   MS100H=ADJSAM*100.
C   MS100S=MSA0HV(WSAM,WSP100)
C
C SET PREAMBLE POINTERS
C
C   IPRBF(PXRECN)=0
C   IPRBF(PXLINO)=IPLIN
C   IPRBF(PXCHAN)=0
C   IPRBF(PXQUAL)=0
C   IPRBF(PXBINT)='INT'
C   IPRBF(PXLBIN)=2
C   IPRBF(PXLCOL)=IPCHIN
C
C CONVERT LOGICAL PPD COLUMNS TO PHYSICAL PPD BINS
C   IPRBF(PXHBIN)=((IPCHAX-IPCHIN+1)*3)+1
C
C   IPRBF(PXHCOL)=IPCHAX
C   IPRBF(PXNOIN)=0
C   IPRBF(PXNODAI)=0
C   IPRBF(PXLJ0I)=0
C   IPRBF(PXHJ0I)=0
C
C INITIALIZE BIN NUMBER TO BIN ONE
C
C   IPBIN=IPRBF(PXLBIN)-1
C
C BUILD SCALE AND BORDER(S)
C   (3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN)

```

```
C
DO 100 NS100=NS100L,NS100H,NS100S
  MSASAM=MS100/100
  IPRTOF(PXBINS+IPBIN*0) = 0
  IPRTOF(PXBINS+IPBIN*1) = 0
  CALL PUTCHR(IPRTOF(PXBINS+IPBIN*2),(1), ':')
  CALL CST4INI(IPRTOF(PXBINS+IPBIN*2),(2),4, MSASAM,4,'0') & ':9999:'
  CALL PUTCHR(IPRTOF(PXBINS+IPBIN*2),(6), ':')
  IPBIN=IPBIN+3
100 CONTINUE
```

```
C
C
C RETURN TO CALLING ROUTINE
```

```
C
900 RETURN
```

C
C
C
C
C
C
C
C
C

INTERNAL
SUBROUTINE RESCRNI & RESAMPLE/SCREEN/COUNT FREQUENCY

```
0 IPRTOF. & PRINT BUFFER
( IPLIN. & PRINT LINE
( IPCHIN. & MINIMUM PRINT COLUMN (LOGICAL)
( IPCHAX. & MAXIMUM PRINT COLUMN (LOGICAL)
-
I MPXBUF. & ARRAY OF MSS PIXEL BUFFERS
( NMIXBF. & NUMBER OF WORDS IN ONE BUFFER
( NXBUFS. & NUMBER OF BUFFERS
I GETBIN) & ROUTINE TO GET BIN VALUE--GETBYT.GETICE.GETINT.GETNUL
```

C
C

C HISTORY

C	D A BECK	LEC	11/07/79	REQUIREMENTS
C	D A BECK	LEC	11/07/79	ALGORITHM DESIGN
C	D A BECK	LEC	11/07/79	ALGORITHM CODING

C
C

C METHOD

```
C
COMPUTE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER
PREAMBLE. FOR EACH SAMPLE, CHECK IF OUTSIDE OF INPUT WINDOW.
CHECK IF OUTSIDE OF RADIANCE LIMITS, AND COUNT FREQUENCY.
NOTE. 3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN.
```

C
C

C MACHINE-DEPENDENT CODE

```
C
NONE.
```

C
C

```

C EXTERNAL REFERENCES
C
C     A4P           & ADJUSTED MSS COORD FOR PRINT/PLOT COORD
C     CST4IN       & CHARACTER STRING FOR INTEGER
C     GETCHR       & GET CHARACTER FROM CHARACTER STRING
C
C EXCEPTIONS
C
C     NONE.
C
C GLOBAL DECLARATIONS
C
C     INCLUDE KOMOHV.LIST      & COMMON OUTPUT WINDOW PACKETS
C     INCLUDE KOMTBL.LIST     & COMMON FREQ/TICK TABLE
C     INCLUDE KOMKLS.LIST     & COMMON CLASSIFICATION INFO
C     INCLUDE PXBDEF.LIST     & DEFINE BUFFER STRUCTURE
C     INCLUDE KOMSYM.LIST     & COMMON SYMBOL TABLE
C     INCLUDE WINDEF.LIST     & DEFINE STRUCTURE OF WINDOW PACKETS
C
C LOCAL DECLARATIONS
C
C     INTEGER MPXBUF(NHIXBF,NXBUFS)      & ARGUMENT
C     INTEGER IPRIBF(1)                 & ARGUMENT
C     INTEGER IPIXL1,IPIXL2,IPIXL3,IPIXL4,IPIXL5,IPIXL6 & VALUE OF PIXEL FOR
C     EACH MPX BUFFER
C     PARAMETER NUMBFS=6
C     INTEGER NBINS0(NUMBFS)            & BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER
C     INTEGER MS100L,MS100H,MS100S      & MSA SAMPLE*100: LOW,HIGH,SPACING
C     INTEGER MSASAM                    & MSA SAMPLE NUMBER
C     INTEGER IPBIN                      & PRINT BUFFER BIN POINTER
C     REAL ADJLIN,ADJSAM                 & ADJUSTED LINE AND SAMPLE
C     INTEGER MS100                      & MSA SAMPLE*100
C     INTEGER KSTPIX                     & CHARACTER STRING OF PIXEL RADIANCE
C
C
C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
C     CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHIN))
C     MS100L=ADJSAM*100.
C     CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHAX))
C     MS100H=ADJSAM*100.
C     MS100S=MSA0HHWMSAM.WSP100)
C
C
C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C
C     DO 150 NUMBUF=1,NXBUFS
C         NBINS0(NUMBUF)=MPXBUF(PXLBIN,NUMBUF)-MPXBUF(PXLSAM,NUMBUF)
C     150 CONTINUE
  
```

```

C
C SET PREAMBLE POINTERS AND BIN POINTER
C
  IPRTBF(PXRECN)=MPXBUF(PXRECN,1)
  IPRTBF(PXLINO)=IPLIN
  IPRTBF(PXCHAN)=0
  IPRTBF(PXQUAL)=0
  IPRTBF(PXBINT)='INT'
  IPRTBF(PXLBIN)=2
  IPRTBF(PXLCOL)=IPCHIN

C
C CONVERT LOGICAL PPD COLUMNS TO PHYSICAL PPD BINS
  IPRTBF(PXNBIN)=((IPCMAX-IPCHIN+1)*3)+1

C
  IPRTBF(PXNCOL)=IPCMAX
  IPRTBF(PXNOIN)=0
  IPRTBF(PXNODA)=0
  IPRTBF(PXLJ01)=0
  IPRTBF(PXHJ01)=0
  IPBIN=IPRTBF(PXLBIN)-1

C
C RESAMPLE/SCREEN RADIANCE/COUNT FREQUENCY
C
  DO 400 MS100=MS100L,MS100M,MS100S
    MSASAM=MS100/100

C RESAMPLE AND SCREEN BUFFER 1
C
  IF ((MSASAM.LT.MPXBUF(PXLSAM,1)).OR.
    & (MSASAM.GT.MPXBUF(PXHSAM,1))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (PIXL1,
    & MPXBUF(PXBINS,1),(MSASAM*NBINS0(1)))
  IF ((PIXL1.GE.MPXBUF(PXNODA,1)) GO TO 350  & NO DATA
    & IF ((PIXL1.LT.LCVLO(1)).OR.
    & ((PIXL1.GT.LCVHI(1))) GO TO 360  & OUT OF RAD LIMITS
    & IF (NLINCH-1.EQ.0) GO TO 320

C BUFFER 2
C
  IF ((MSASAM.LT.MPXBUF(PXLSAM,2)).OR.
    & (MSASAM.GT.MPXBUF(PXHSAM,2))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (PIXL2,
    & MPXBUF(PXBINS,2),(MSASAM*NBINS0(2)))
  IF ((PIXL2.LT.LCVLO(2)).OR.
    & ((PIXL2.GT.LCVHI(2))) GO TO 360  & OUT OF RAD LIMITS
    & IF (NLINCH-2.EQ.0) GO TO 290

C BUFFER 3
C
  IF ((MSASAM.LT.MPXBUF(PXLSAM,3)).OR.
    & (MSASAM.GT.MPXBUF(PXHSAM,3))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (PIXL3,
    & MPXBUF(PXBINS,3),(MSASAM*NBINS0(3)))
  IF ((PIXL3.LT.LCVLO(3)).OR.
    & ((PIXL3.GT.LCVHI(3))) GO TO 360  & OUT OF RAD LIMITS
  
```

IF (NLINCH-3.EQ.0) GO TO 260

C
 C BUFFER 4
 C

IF ((MSASAM.LT.MPKBUF(PXLSAM.4)).OR.
 (MSASAM.GT.MPKBUF(PXHSAM.4))) GO TO 350 8 SAMPLE NOT IN BUFFER
 CALL GETBIN (PIXL4.
 MPKBUF(PXBINS.4).(MSASAM+NBINS0(4)))
 IF ((PIXL4.LT.LCVLO(4)).OR.
 (PIXL4.GT.LCVHI(4))) GO TO 360 8 OUT OF RAD LIMITS
 IF (NLINCH-4.EQ.0) GO TO 230

C
 C BUFFER 5
 C

IF ((MSASAM.LT.MPKBUF(PXLSAM.5)).OR.
 (MSASAM.GT.MPKBUF(PXHSAM.5))) GO TO 350 8 SAMPLE NOT IN BUFFER
 CALL GETBIN (PIXL5.
 MPKBUF(PXBINS.5).(MSASAM+NBINS0(5)))
 IF ((PIXL5.LT.LCVLO(5)).OR.
 (PIXL5.GT.LCVHI(5))) GO TO 360 8 OUT OF RAD LIMITS
 IF (NLINCH-5.EQ.0) GO TO 200

C
 C BUFFER 6
 C

IF ((MSASAM.LT.MPKBUF(PXLSAM.6)).OR.
 (MSASAM.GT.MPKBUF(PXHSAM.6))) GO TO 350 8 SAMPLE NOT IN BUFFER
 CALL GETBIN (PIXL6.
 MPKBUF(PXBINS.6).(MSASAM+NBINS0(6)))
 IF ((PIXL6.LT.LCVLO(6)).OR.
 (PIXL6.GT.LCVHI(6))) GO TO 360 8 OUT OF RAD LIMITS

C
 C COUNT FREQUENCY
 C

200 NFREQ(PIXL6+1.6)-NFREQ(PIXL6+1.6)+1
 230 NFREQ(PIXL5+1.5)-NFREQ(PIXL5+1.5)+1
 260 NFREQ(PIXL4+1.4)-NFREQ(PIXL4+1.4)+1
 290 NFREQ(PIXL3+1.3)-NFREQ(PIXL3+1.3)+1
 320 NFREQ(PIXL2+1.2)-NFREQ(PIXL2+1.2)+1
 350 NFREQ(PIXL1+1.1)-NFREQ(PIXL1+1.1)+1

C
 C INSERT RADIANCE VALUE INTO PRINT BUFFER
 C (3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN)
 C

CALL CSTNIN(KSTPIX.(1).3.
 PIXL1.3.'0')
 CALL GETCHR(IPRTBF(PXBINS-IPBIN+0). 8 HUNDREDS DIGIT
 KSTPIX.(1))
 CALL GETCHR(IPRTBF(PXBINS-IPBIN+1). 8 TENS DIGIT
 KSTPIX.(2))
 CALL GETCHR(IPRTBF(PXBINS-IPBIN+2). 8 UNITS DIGIT
 KSTPIX.(3))
 IF(IPRTBF(PXBINS-IPBIN+0).EQ.'0') IPRTBF(PXBINS-IPBIN+0)='.'
 GO TO 390

C
 C

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICL13
011

C INSERT 'NO DATA' SYMBOLS INTO PRINT BUFFER

C

```
350      |PRTF(PXBINS+IPBIN+0)=':'  
        |PRTF(PXBINS+IPBIN+1)=':'  
        |PRTF(PXBINS+IPBIN+2)=':'  
        GO TO 390
```

C

C

C INSERT 'NO INFO' SYMBOLS INTO PRINT BUFFER

C

```
360      |PRTF(PXBINS+IPBIN+0)=':'  
        |PRTF(PXBINS+IPBIN+1)=':'  
        |PRTF(PXBINS+IPBIN+2)=':'  
        IPBIN=IPBIN+3
```

390 CONTINUE

C

C

C RETURN TO CALLING ROUTINE

C

```
900 RETURN  
    END
```

ORIGINAL PAGE IS
OF POOR QUALITY

U-3


```
      SUBROUTINE PICL14  @ LIST GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)
-----
C
C (E H SCHLOSSER)
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
-----
C
C      MNOTE
C      NVIATO
C
C
C      EXTERNAL PIC129.PIC119
C      CALL TRACE
C
C
C
C      CALL MNOTE(
C      - 'LIST.GRADIENT/LAPLACIAN/VARIANCE NOT YET IMPLEMENTED')
C      CALL NVIATO(  PIC129.PIC119)
C      RETURN
C      END
```

```
      SUBROUTINE PICL18  & LIST CLASS (PHASE 5)
      -----
C
C (E H SCHLOSSER)
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
C -----
C
C      HDNOTE
C      NVIATO
C
C
C      EXTERNAL PIC129.PICL19
C      CALL TRACE
C
C
C
C      CALL HDNOTE('LIST.CLASS NOT YET IMPLEMENTED')
C      CALL NVIATO(PIC129.PICL19)
C      RETURN
C      END
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICL19
002

C ANY DIAGNOSTICS???

C

900 IF(INDFATL.EQ.0) GO TO 920

CALL MDNOTE('FATAL ERRORS -- NO LIST GENERATED')
GO TO 990

920 IF(INDWARN.EQ.0) GO TO 980

CALL MDNOTE('PREVIOUS WARNINGS -- NO LIST GENERATED')
IF(INDATCH.EQ.0) WRITE(6,925)

925 FORMAT(4X,'**TRY AGAIN!')

CALL MDCLRMI(NULCST)
GO TO 990

C

C

C PREPARE FOR NEXT WINDOW

C

960 NWINDOW=NWINDOW+1

C

C

C RETURN TO NEXT STATEMENT IN CALLING ROUTINE

C

990 RETURN
END

ORIGINAL PAGE IS
OF POOR QUALITY

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICPAR
002

```
300 CALL GETSIN(ITEMP,+1,-1,'EXTRA PARTITION SPECIFICATION ---')
C
C
C CHECK RADIANCE LIMITS
C
      IF(LCVLO1.GT.LCVHI1) CALL MOWARN('NO RADIANCE LIMITS')
      IF(MDATAC.NE.0) GO TO 850      & DATA/CHECKOUT MODE
C
C CALIBRATE SPACING/WINDOW
C
      CALL CALSPA
      CALL CALWIN(0.)
C
C
C CROP OUTPUT WINDOW TO FIT INPUT WINDOW
C
      CALL CROPWIN(2500,3500)
C
C
C CHECK FOR DIAGNOSTICS
C
      IF(INDTOTL.NE.0) GO TO 800
C
C
C CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
      IF(NWINDOW.LT.0) CALL OPRPIC      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
      NWINDOW=ABS(NWINDOW)
      NPAGE=0
C
C
C PRINT WINDOW HEADING FOR UNIT 8
C
      IF(MBATCH.NE.0) CALL MOUNIT(4,8)
      WRITE(8,415) NWINDOW,MTERRAL
415 FORMAT(' WINDOW NUMBER ',J3,8X,'PARTITION',8X,4A6)
      CALL IDLU3(8)
      CALL IDERT(8)
      CALL IOCPIC(8)
C
C
C PRINT WINDOW HEADING FOR UNIT 10
C
450 CALL MOUNIT(4,10)
      WRITE(10,415) NWINDOW,MTERRAL
      CALL IDLU3(10)
      CALL IDERT(10)
      CALL IOCPIC(10)
      GO TO 900
C
C
C WARNINGS ENCOUNTERED -- NEXT CALL IS TO PICPA9
C
800 CALL NVIATO(PIC129,PICPA9)
      KLSTYP=0      & NO PARTITION GENERATED
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICPAR
003

```
      00 TO 900  
C  
C  
C DATA/CHECKOUT MODE -- GET NEXT COMMAND  
C  
C 050 CALL NVIATO(PIC000.NULSUB)  
C  
C  
C  
C RETURN FOR CALL TO NAMED SUBROUTINE  
C  
C 900 KOND=  
      RETURN  
      END
```


SUBROUTINE PICPA4 8 PARTITION BY GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)

```
C
C
C (E H SCHLOSSER)
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
C -----
C
C      HDNOTE
C      NVIATO
C
C      EXTERNAL PIC129.PICPA9
C      CALL TRACE
C
C
C      CALL HDNOTE(
C      * 'PARTITION.GRADIENT/LAPLACIAN/VARIANCE NOT YET IMPLEMENTED')
C      CALL NVIATO(PIC129.PICPA9)
C      RETURN
C      END
```


DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICPAB
001

```
      SUBROUTINE PICPAB 0 PARTITION FACTOR 0PAGE (PHASE 0)  
C -----  
C  
C (E H SCHLOSSER)  
C  
C  
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED  
C -----  
C  
C      MNOTE  
C      NVIATO  
C  
C  
C      EXTERNAL PIC000.NULSUB  
C      CALL TRACE  
C  
C  
C  
C      CALL MNOTE('PARTITION COMMAND NOT YET IMPLEMENTED')  
C      CALL NVIATO(PIC000.NULSUB)  
C      RETURN  
C      END
```


C
 C 2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE PICTURE.
 C
 C 3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS

COND	ACTION	DIAGNOSTIC
PROCESSING DEFAULT COMMANDS (NWINDOW=0)	NONE	WARNING
KLSTYP IN COMMON UNDEFINED	KLSTYP='RAD'	NONE
KLSTYP SPECIFICATION MISSING	USE COMMON KLSTYP	NONE
KLSTYP SPECIFICATION INVALID	NONE	WARNING
EXTRA SPECIFICATION	NONE	WARNING
LIMIT CHANNEL VALUE RANGE IS NULL (LCVLOI>LCVHI)	NONE	WARNING
COLOR MODE SWITCH NOT ON	NONE	WARNING
BATCH RUN	NONE	WARNING
DATA/CHECKOUT MODE	'TO' ROUTINE IS NULSUB	NONE
WARNING(S) OR FATAL ERROR(S)	'TO' ROUTINE IS PICPI9	NONE

C
 C GLOBAL DECLARATIONS
 C -----
 C

INCLUDE KOMXOT.LIST	% COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST	% COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST	% COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMTBL.LIST	% COMMON TICK/FREQ/FACTOR TABLE
INCLUDE KOMSYM.LIST	% COMMON SYMBOL TABLE
INCLUDE KOMKS.LIST	% COMMON COLOR SCREEN PARAMETERS
INCLUDE WINDEF.LIST	% DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMOWH.LIST	% COMMON OUTPUT WINDOW PACKETS
INCLUDE NULCST.LIST	% DEFINE NULL CHARACTER STRING

C
 C LOCAL DECLARATIONS
 C -----
 C

INTEGER KLSAVE	% SAVE LOCATION FOR KLSTYP
INTEGER INTMP	% TEMPORARY
INTEGER NPRLIN,NPRCOL	% NUMBER OF PPD LINES/COLUMNS IN PICTURE

C
 C PROCEDURE
 C -----
 C

CALL TRACE

C
 C CHECK IF COMMAND IS LEGAL
 C

```

IF(NWINDOW.EQ.0) CALL MDWARN('INVALID DEFAULT COMMAND')
IF(BATCH.NE.0) CALL MDWARN('PICTURE COMMAND NOT ALLOWED IN BATCH')
IF(COLOR.EQ.0) CALL MDWARN(

```

* 'PICTURE COMMAND NOT ALLOWED (COLOR MODE NOT ON)')

C
 C
 C
 C

```

KTBLTY=' NUL'      & MARK OLD FREQ TABLE AS DESTROYED
KLSAVE=KLSTYP      & SAVE PREVIOUS PICTURE TYPE
IF((KLSTYP.NE.'GRA').AND.
& (KLSTYP.NE.'LAP').AND.
& (KLSTYP.NE.'VAR').AND.
& (KLSTYP.NE.'CLA'))      & IF UNDEFINED ...
& KLSTYP='RAD'          & ... THEN MAKE IT RADIANCE! ...
CALL GETSKH(KLSTYP.(3), NULCST) & ... UNLESS SPECIFIED BY PICTURE! CMD
  
```

C
 C
 C
 C

C CHECK PICTURE TYPE

```

IF(KLSTYP.NE.'RAD') GO TO 240      & RADIANCE!?
CALL NVIATO( PIC345.PICP13) & NEXT CALL IS TO PICP13
GO TO 300
240 IF((KLSTYP.NE.'GRA').AND.      & GRADIENT!?
& (KLSTYP.NE.'LAP').AND.        & LAPLACIAN!?
& (KLSTYP.NE.'VAR')) GO TO 250  & VARIANCE!?
CALL NVIATO( PIC345.PICP14) & NEXT CALL IS TO PICP14
GO TO 300
250 IF(KLSTYP.NE.'CLA') GO TO 280  & CLASS!?
CALL NVIATO( PIC345.PICP15) & NEXT CALL IS TO PICP15
GO TO 300
280 CALL WARN( 'BAD PICTURE TYPE --')
KLSAVE=KLSTYP      & RESTORE PREVIOUS PICTURE TYPE
  
```

C
 C
 C
 C

C DRAIN SPECS FOR CURRENT COMMAND

```

300 CALL GETSIN(TEMP. +1.-1.'EXTRA PICTURE SPECIFICATION --')
  
```

C
 C
 C
 C

C CHECK RADIANCE LIMITS

```

IF(LCVLOI.GT.LCVHI) CALL MOWARN( 'NO RADIANCE LIMITS')
IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE
  
```

C
 C
 C
 C

C CALIBRATE CHANNELS/COLORS/SPACING/WINDOW

```

CALL CALCHA
CALL CALCOL
CALL CALSPA
CALL CALWIN( 0.)
  
```

C
 C
 C
 C

C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER

```

IF(NDTOTL.NE.0) GO TO 900
IF(NWINDOW.LT.0) CALL OPRPIC      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
NWINDOW=ABS(NWINDOW)
  
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICPIC
004

```
      NPAGE=0
C
C
C CROP OUTPUT WINDOW TO FIT INPUT WINDOW & COLOR SCREEN
C
      CALL CROPON( KSLINE,KSCOLM)
      IF (NDTOTL.NE.0) GO TO 900
C
C
C COMPUTE SIZE OF PRINT/PLOT DEVICE (COLOR SCREEN) WINDOW
C
      NPRLN=IFIX(PPDOWN(WLIN.WMAX))-IFIX(PPDOWN(WLIN.WMIN))+1
      NPRCOL=IFIX(PPDOWN(WCOL.WMAX))-IFIX(PPDOWN(WCOL.WMIN))+1
C
C
C CHECK FOR DIAGNOSTICS
C
      IF(NDTOTL.NE.0) GO TO 900
C
C
C CLEAR SCREENS AND PRINT WINDOW HEADING
C
      CALL EAPRNT(0.1,KSON)      & ROUTE UNIT 6 OUTPUT TO COLOR SCREEN
      CALL EAPRNT(0.1,KSCLER)   & CLEAR COLOR SCREEN
      WRITE(6,415) NWNDOW,MTERRAL
415 FORMAT('> WINDOW NUMBER ',J3.6X,'PICTURE',6X.4A6)
      CALL EAPRNT(0.1,KSOFF)     & ROUTE UNIT 6 OUTPUT TO B&W SCREEN
      WRITE(6,415) NWNDOW,MTERRAL
      CALL IDLU3( 6)
      CALL IDERT( 6)
      CALL IDCPI( 6)
C
C
C ANY DIAGNOSTICS???
C
900 IF(NDTOTL.EQ.0) GO TO 990
      IF(MDATAC.NE.0) CALL NVIATO( PIC000.NULSUB)      & DATA/CHECKOUT
      IF(MDATAC.EQ.0) CALL NVIATO( PIC129.PICP19)
C
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KOMD=' '
      RETURN
      END
```


C
 C GLOBAL DECLARATIONS
 C -----
 C

```

INCLUDE KONXQT.LIST      % COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONKLS.LIST     % COMMON CLASSIFICATION INFO
INCLUDE KONSYH.LIST     % COMMON SYMBOL TABLE
INCLUDE KONKS.LIST      % COMMON COLOR SCREEN PARAMETERS
INCLUDE WINDEF.LIST     % DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KONOWH.LIST     % COMMON OUTPUT WINDOW PACKETS
INCLUDE KONTBL.LIST     % COMMON TICK/FREQ TABLE
INCLUDE PICDEF.LIST     % DEFINE PICTAB PARAMETERS
INCLUDE PXBDEF.LIST     % DEFINE PIXEL BUFFER STRUCTURE
INCLUDE MAXINT.LIST     % MAXIMUM INTEGER VALUE
  
```

C
 C LOCAL DECLARATIONS
 C -----
 C

```

PARAMETER NXBUFS=6      % # OF MSS PIXEL BUFFERS IN ARRAY
INTEGERS IN MSA BUF = #INTS PREAMBLE + (#BINS+3)/4 + (#EXTRA BYTES+3)/4
PARAMETER NWIXBF = (PXBINS-1) + (3548+3)/4 + (19+3)/4
%WDS COLOR/INTENSITY BUF=WDS PREAM+256
PARAMETER NWIKBF=(PXBINS-1) + 256
PARAMETER NFRQCH=6      % # WDS PER CHANNEL IN LOCAL FREQ TABLE
PARAMETER NFRQSZ=128    % # CHANNELS IN LOCAL FREQ TABLE
INTEGER MPXBUF(NWIXBF,NXBUFS) % ARRAY OF MSS PIXEL BUFFERS
INTEGER KIBUF(NWIKBF)   % COLOR/INTENSITY BUFFER
INTEGER NFRQ(NFRQSZ,NFRQCH) % LOCAL FREQUENCY TABLE (SCOPE INCLUDES
% INTERNAL ROUTINE RESCOL)
INTEGER NFRQCR(10,15)  % LOCAL CROSS FREQUENCY TABLE
INTEGER IPLIN          % PPD LINE
INTEGER IPCHIN,IPCHAX % MINIMUM AND MAXIMUM PPD COLUMN
INTEGER IPLMIN,IPLMAX % MINIMUM AND MAXIMUM PPD LINE
REAL ADJLIN,ADJSAM     % ADJUSTED LINE AND SAMPLE
INTEGER ML100L,ML100H,ML100S % MSA LINE*100: LOW,HIGH,SPACING
INTEGER MSALIN        % MSA LINE NUMBER
INTEGER MSASLO,MSASHI % LOW AND HIGH MSA SAMPLE
INTEGER ISTAT         % I/O STATUS
INTEGER NPRLIN,NPRCOL % NUMBER OF PPD LINES AND COLUMNS
INTEGER LASTLN       % LAST SCAN LINE READ
  
```

C
 C PROCEDURE
 C -----
 C

CALL TRACE

C
 C INITIALIZE MINIMUM AND MAXIMUM PPD LINES AND COLUMNS
 C

```

IPLMIN=PPDOWN(WLIN,WMIN)
IPLMAX=PPDOWN(WLIN,WMAX)
IPCHIN=PPDOWN(WCOL,WMIN)
IPCHAX=PPDOWN(WCOL,WMAX)
  
```

C

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICP13
003

```
C
C COMPUTE NUMBER OF PPD LINES AND COLUMNS
C
      NPRLIN=IPLMAX-IPLMIN+1
      NPRCOL=IPCHAX-IPCHIN+1
C
C CLEAR LOCAL FREQUENCY TABLES
C
      DO 250 I=1,NLIMCH
          DO 200 K=1,NFRQSZ
              NFREQ(K,I)=0
200      CONTINUE
250      CONTINUE
          DO 280 I=1,10
              DO 270 K=1,15
                  NFCRO(I,K)=0
270      CONTINUE
280      CONTINUE
C
C INITIALIZE LOW AND HIGH LINES AND SPACING
C
      CALL A4P (ADJLIN,ADJSAM,   FLOAT(IPLMIN),1.)
      ML100L=ADJLIN*100.
      CALL A4P (ADJLIN,ADJSAM,   FLOAT(IPLMAX),1.)
      ML100H=ADJLIN*100.
      ML100S=MSAONW*(MLIN,MSP100)
C
C POSITION AT TOP OF WINDOW
C
      MSALIN=ML100L/100
      CALL GETRAD (MPXBUF,(0),(NXBUFS),ISTAT,   MSALIN,0.0)
      IF (ISTAT.NE.'BADF') GO TO 300
          CALL HDFATL ('BADF (BAD FILE) ON UNIT 3')
          GO TO 900
300      CONTINUE
C
C TURN ON COLOR SCREEN
C
      CALL EAPRNT(0.1,KSON)
C
C READ, MASK, RESAMPLE, SCREEN, SYMBOLIZE AND WRITE SCAN LINES
C
      IPLIN=IPLMIN
      LASTLN=-MAXINT
      DO 350 ML100=ML100L,ML100H,ML100S
          MSALIN=ML100/100
          CALL A4P (ADJLIN,ADJSAM,   FLOAT(IPLIN),FLOAT(IPCHIN))
          MSASLO=ADJSAM
          CALL A4P (ADJLIN,ADJSAM,   FLOAT(IPLIN),FLOAT(IPCHAX))
          MSASHI=ADJSAM
          IF (MSALIN.NE.LASTLN) CALL GETRAD (MPXBUF,(NHIXBF),(NXBUFS),
```

```

      6          ISTAT.  MSALIN,MSASLO,MSASHI)
      LASTLN=MSALIN
      IF ((ISTAT.NE.'BADF').AND.(ISTAT.NE.'OFL')) GO TO 320
      CALL HDFATL (COS4CS(ISTAT,1,4),
      ' WHILE READING ON UNIT 3')
      6          GO TO 900
320    CALL MSKPIX (MPXBUF(1,1),  MPXBUF(1,1))
      IF ((MPXBUF(PXBINT,1).EQ.'BYT').OR.
      (MPXBUF(PXBINT,1).EQ.'CHR').OR.
      (MPXBUF(PXBINT,1).EQ.'INT').OR.
      (MPXBUF(PXBINT,1).EQ.'NUL')) GO TO 330
      CALL HDFATL('INVALID BIN TYPE ',
      COS4CS(MPXBUF(PXBINT,1),1,3),' IN PICP13')
      6          GO TO 900
330    IF (MPXBUF(PXBINT,1).EQ.'BYT') CALL RESCOL (KIBUF,(IPLIN),
      (IPCHIN),(IPCHAX),  MPXBUF,(NHIXBF),(NXBUFS),GETBYT)
      IF (MPXBUF(PXBINT,1).EQ.'CHR') CALL RESCOL (KIBUF,(IPLIN),
      (IPCHIN),(IPCHAX),  MPXBUF,(NHIXBF),(NXBUFS),GETICE)
      IF (MPXBUF(PXBINT,1).EQ.'INT') CALL RESCOL (KIBUF,(IPLIN),
      (IPCHIN),(IPCHAX),  MPXBUF,(NHIXBF),(NXBUFS),GETINT)
      IF (MPXBUF(PXBINT,1).EQ.'NUL') CALL RESCOL (KIBUF,(IPLIN),
      (IPCHIN),(IPCHAX),  MPXBUF,(NHIXBF),(NXBUFS),GETNUL)
      CALL EAPRNT(1,NH4NB(KIBUF(PXBIN)),KIBUF(PXBINS))
      IPLIN=IPLIN+1
350 CONTINUE
C
C
C MOVE DATA FROM LOCAL FREQ TABLES TO COMMON TABLE (REPLACING TICKS)
C
      KTOLTY='FREQ'
      KTBLNH=NHNDOW
      DO 600 I=1,NLINCH
      DO 550 K=1,NFRQSZ
      KFREQ(K,1)=NFREQ(K,1)
550    CONTINUE
800 CONTINUE
      DO 700 I=1,10
      DO 650 K=1,15
      KFCRO(I,K)=NFCRO(I,K)
650    CONTINUE
700 CONTINUE
C
C
C TURN OFF COLOR SCREEN AND RETURN -- NEXT CALL IS TO PICP19
C
900 CALL EAPRNT(0,1,KSOFF)
      CALL NVIATO ( PIC129,PICP19)
      RETURN
C
C
C
C
C
C
C
C
C
C

```

```

C      INTERNAL
      SUBROUTINE RESCOL( 3 RESAMPLE/SCREEN/COUNT FREQUENCY/COLOR
0 KIBUF.      3 COLOR/INTENSITY BUFFER
( IPLIN.      3 PPD LINE
( IPCHIN.     3 MINIMUM PPD COLUMN
( IPCMAX.     3 MAXIMUM PPD COLUMN
-
1 MPXBUF.     3 ARRAY OF MSS PIXEL BUFFERS
( NWIXBF.     3 NUMBER OF WORDS IN ONE PIXEL BUFFER
( NXBUFS.     3 NUMBER OF PIXEL BUFFERS
1 GETBIN)     3 ROUTINE TO GET BIN VALUE--GETBYT.GETICE.GETINT.GETNUL

C
C
C METHOD
C
C COMPUTE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER PREAMBLE.
C FOR EACH SAMPLE. CHECK IF OUTSIDE OF INPUT WINDOW. CHECK IF OUT-
C SIDE OF RADIANCE LIMITS. COUNT FREQUENCY. AND SYMBOLIZE. INSERT
C TICKS.
C
C
C EXTERNAL REFERENCES
C
C A4P          3 ADJUSTED MSS COORD FOR PRINT/PLOT COORD
C GETICE       3 GET INTEGER-CHARACTER-EQUIVALENT FROM CHARACTER STRING
C PUTBYT       3 PUT BYTE INTO BYTE STRING
C MOVBST       3 MOVE BYTE STRING
C
C
C GLOBAL DECLARATIONS
C
C INCLUDE KOMOWH.LIST      3 COMMON OUTPUT WINDOW PACKETS
C INCLUDE KOMTBL.LIST     3 COMMON FREQ/TICK TABLE
C INCLUDE KOMKLS.LIST     3 COMMON CLASSIFICATION INFO
C INCLUDE PXBDEF.LIST     3 DEFINE PIXEL BUFFER STRUCTURE
C INCLUDE KOMSYH.LIST     3 COMMON SYMBOL TABLE
C INCLUDE KOMKS.LIST      3 COMMON COLOR SCREEN PARAMETERS
C INCLUDE WINDEF.LIST     3 DEFINE WINDOW PACKETS
C INCLUDE MAXINT.LIST     3 DEFINE MAXIMUM INTEGER
C
C
C LOCAL DECLARATIONS
C
      PARAMETER NUMBFS=6      3 NUMBER OF MPX BUFFERS
      INTEGER MPXBUF(NWIXBF,NXBUFS)  3 ARGUMENT
      INTEGER KIBUF(1)        3 ARGUMENT
      INTEGER NBINSO(NUMBFS)   3 BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER
      INTEGER MS100L,MS100M,MS100S  3 MSA SAMPLE*100: LOW,HIGH,SPACING
      INTEGER MSASAM           3 MSA SAMPLE NUMBER
      INTEGER KIBIN           3 COLOR/INTENSITY BUFFER BIN POINTER
      REAL ADJSAM             3 ADJUSTED SAMPLE NUMBER
      INTEGER IPIXL1,IPIXL2,IPIXL3,
      IPIXL4,IPIXL5,IPIXL6      3 PIXEL VALUE FOR EACH BUFFER

C
      INTEGER IKENU           3 INTEGER-COLOR-EQUIVALENT (NEW)
      INTEGER KIKE,KIKENU     3 CODED INTEGER-COLOR-EQUIVALENT (CURRENT),NEW)
  
```

```

      INTEGER IIE, IIEHU      8      INTEGER-INTENSITY-EQUIVALENT (CURRENT,NEW)
      INTEGER IIE            8 CODED INTEGER-INTENSITY-EQUIVALENT (CURRENT)
C
C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
      CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHIN))
      NS100L=ADJSAM*100.
      CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHAX))
      NS100H=ADJSAM*100.
      NS100S=MSAOWH(MSAM,WSP100)
C
C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C
      DO 150 NUMBUF=1,NXBUFS
          NB(NS0(NUMBUF))=MPXBUF(PXLBIN,NUMBUF)-MPXBUF(PXLSAM,NUMBUF)
150 CONTINUE
C
C INITIALIZE PREAMBLE FOR COLOR-INTENSITY BUFFER
C
      KIBUF(PXRECNI)=MPXBUF(PXRECNI,1)
      KIBUF(PXLINO)=IPLIN
      KIBUF(PXCHAN)=0
      KIBUF(PXQUAL)=0
      KIBUF(PXBINT)='BYT'
      KIBUF(PXLBIN)=1
      KIBUF(PXLCOL)=IPCHIN
      KIBUF(PXHCOL)=IPCHAX
      KIBUF(PXNOIN)=0
      KIBUF(PXNODA)=0
      KIBUF(PXLJOI)=0
      KIBUF(PXHJOI)=0
C
C INITIALIZE BIN POINTER & FIRST BIN OF COLOR-INTENSITY BUFFER
C
      KIBIN=KIBUF(PXLBIN)
      CALL PUTBYT(KIBUF(PXBINS),(KIBIN), 33)      8 ASCII: 1
          '1' TURNS ON PICTURING IN NORTHSTAR/ISC
          <CR> <LF> AT END OF EACH LINE TURN OFF PICTURING
C
C INITIALIZE CURRENT INTEGER-INTENSITY-EQUIV & CODED INTEGER-COLOR-EQUIV
C
      IIE=MAXINT
      IIEH=MAXINT
C
C RESAMPLE/SCREEN RADIANCE/COUNT FREQUENCY/LOOK UP SYMBOLS
C
      DO 460 NS100=NS100L,NS100H,NS100S
          M\ASAM=NS100/100
  
```

DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

PICP13
 007

```

C
C BUFFER 1
C
      IF ((MSASAM.LT.MPXBUF(PXLSAM.1)).OR.
          (MSASAM.GT.MPXBUF(PXHSAM.1))) GO TO 370      & SAMPLE NOT IN BUFFER
      CALL GETBIN ((PIXL1.
          -      MPXBUF(PXBINS.1).(MSASAM+NBINS0(1)))
      (PIXL2=PIXL1      & SECOND CHANNEL VALUE SAME AS FIRST IF UNDEFINED
      IF ((PIXL1.GE.MPXBUF(PXNODA.1)) GO TO 350      & NO DATA
          IF ((PIXL1.LT.LCVLO1).OR.
          &      ((PIXL1.GT.LCVHI1))) GO TO 360      & OUT OF RAD LIMITS
          IF (NLINCH-1.EQ.0) GO TO 320
  
```

```

C
C BUFFER 2
C
      IF ((MSASAM.LT.MPXBUF(PXLSAM.2)).OR.
          (MSASAM.GT.MPXBUF(PXHSAM.2))) GO TO 350      & SAMPLE NOT IN BUFFER
      CALL GETBIN ((PIXL2.
          -      MPXBUF(PXBINS.2).(MSASAM+NBINS0(2)))
      IF ((PIXL2.LT.LCVLO(2)).OR.
          &      ((PIXL2.GT.LCVHI(2))) GO TO 360      & OUT OF RAD LIMITS
          IF (NLINCH-2.EQ.0) GO TO 290
  
```

```

C
C BUFFER 3
C
      IF ((MSASAM.LT.MPXBUF(PXLSAM.3)).OR.
          (MSASAM.GT.MPXBUF(PXHSAM.3))) GO TO 350      & SAMPLE NOT IN BUFFER
      CALL GETBIN ((PIXL3.
          -      MPXBUF(PXBINS.3).(MSASAM+NBINS0(3)))
      IF ((PIXL3.LT.LCVLO(3)).OR.
          &      ((PIXL3.GT.LCVHI(3))) GO TO 360      & OUT OF RAD LIMITS
          IF (NLINCH-3.EQ.0) GO TO 260
  
```

```

C
C BUFFER 4
C
      IF ((MSASAM.LT.MPXBUF(PXLSAM.4)).OR.
          (MSASAM.GT.MPXBUF(PXHSAM.4))) GO TO 350      & SAMPLE NOT IN BUFFER
      CALL GETBIN ((PIXL4.
          -      MPXBUF(PXBINS.4).(MSASAM+NBINS0(4)))
      IF ((PIXL4.LT.LCVLO(4)).OR.
          &      ((PIXL4.GT.LCVHI(4))) GO TO 360      & OUT OF RAD LIMITS
          IF (NLINCH-4.EQ.0) GO TO 230
  
```

```

C
C BUFFER 5
C
      IF ((MSASAM.LT.MPXBUF(PXLSAM.5)).OR.
          (MSASAM.GT.MPXBUF(PXHSAM.5))) GO TO 350      & SAMPLE NOT IN BUFFER
      CALL GETBIN ((PIXL5.
          -      MPXBUF(PXBINS.5).(MSASAM+NBINS0(5)))
      IF ((PIXL5.LT.LCVLO(5)).OR.
          &      ((PIXL5.GT.LCVHI(5))) GO TO 360      & OUT OF RAD LIMITS
          IF (NLINCH-5.EQ.0) GO TO 200
  
```

```

C
C BUFFER 6
C
      IF ((MSASAM.LT.MPXBUF(PXLSAM.6)).OR.
  
```

DAH PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

PICP13
 000

```

      6      (MSASAM.07.NPKBUF(PKMSAM.6)) GO TO 350      6 SAMPLE NOT IN BUFFER
      CALL GETBIN (IPIXL6.
      *      NPKBUF(PKBINS.6).(MSASAM+NBINS0(6)))
      *      IF ((IPIXL6.LT.LCVLO(6)).OR.
      6      (IPIXL6.07.LCVHI(6))) GO TO 360      6 OUT OF RAD LIMITS
C
C COUNT FREQUENCY
C
      NFREQ(IPIXL6+1.6)=NFREQ(IPIXL6+1.6)+1
      200      NFREQ(IPIXL5+1.5)=NFREQ(IPIXL5+1.5)+1
      230      NFREQ(IPIXL4+1.4)=NFREQ(IPIXL4+1.4)+1
      260      NFREQ(IPIXL3+1.3)=NFREQ(IPIXL3+1.3)+1
      290      NFREQ(IPIXL2+1.2)=NFREQ(IPIXL2+1.2)+1
      320      NFREQ(IPIXL1+1.1)=NFREQ(IPIXL1+1.1)+1
C
C
C INFO -- LOOK UP NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
C
      CALL GETICE(IIEINU.
      *      KSYN(IPIXL1+1).(5))
      *      CALL GETICE(IKENU.
      *      KSYN(IPIXL2+1).(6))
      *      GO TO 370
C
C
C NO DATA -- ASSIGN NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
C
      350      CALL GETICE(IIEINU.
      *      KSYN(ISYMNO+1).(5))
      *      CALL GETICE(IKENU.
      *      KSYN(ISYMNO+1).(6))
      *      GO TO 370
C
C
C NO INFO -- ASSIGN NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
C
      360      CALL GETICE(IIEINU.
      *      KSYN(ISYMI+1).(5))
      *      CALL GETICE(IKENU.
      *      KSYN(ISYMI+1).(6))
C
C
C COUNT INTENSITY X COLOR CROSS FREQUENCY
C
      370      NFRERO(IIEINU+1.(KENU+1))=NFRERO(IIEINU+1.(KENU+1))+1
C
C
C IF NEW CODED I-K-E. PUT CODED I-K-E & CODED I-I-E IN BUFFER
C
      KIKENU=KSKIKE(IKENU+1)
      IF(KIKENU.EQ.KIKE) GO TO 380
      KIKE=KIKENU
      KIBIN=KIBIN+1
      CALL PUTBYT(KIQUF(PKBINS1).(KIBIN1. KIKE)
      *      IIE=IIEINU
      *      KIE=IIE+57
  
```

DM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICP13
000

```

      KIBIN=KIBIN+1
      CALL PUTBYT(KIBUF(PXBINS),(KIBIN), KIE)
      GO TO 400
C
C
C SAME COLOR -- IF SAME INTENSITY & BIN. UPDATE CODED IIE IN BUFFER
C
300   IF(IIEU.NE.IIE) GO TO 300
      IF(KIE.GT.II) GO TO 300
      KIE=KIE+10
      CALL PUTBYT(KIBUF(PXBINS),KIBIN), KIE)
      GO TO 400
C
C
C SAME COLOR WITH NEW INTENSITY OR NEW BIN
C
300   IIE=IIEU
      KIE=KIE+97
      KIBIN=KIBIN+1
      CALL PUTBYT(KIBUF(PXBINS),(KIBIN), KIE)
C
C
C LOOP TO PROCESS NEXT PIXEL
C
400 CONTINUE
C
C
C STORE POINTER TO LAST BIN IN PREAMBLE & PAD BUFFER WITH 8 NUL'S
C
      KIBUF(PXBIN)=KIBIN
      CALL MOVBS(KIBUF(PXBINS),(KIBIN+1),(8),
      .      0,(1),(1),0)
C
C
600 RETURN
      END
```



```
      SUBROUTINE PICPI4  0 PICTURE GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)  
-----  
C  
C (E H SCHLOSSER)  
C  
C  
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED  
C-----  
C  
C      MONOTE  
C      NVIATO  
C  
C  
C      EXTERNAL PIC129.PICPI9  
C      CALL TRACE  
C  
C  
C  
C      CALL MONOTE!  
C      - 'PICTURE.GRADIENT/LAPLACIAN/VARIANCE NOT YET IMPLEMENTED'  
C      CALL NVIATO!  PIC129.PICPI9!  
C      RETURN  
C      END
```

ORIGINAL PAGE 13
OF 13 PAGES

SUBROUTINE PICP19 3 PICTURE SCAN DATA (PHASE 9)

C
C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 05/17/79 REQUIREMENTS & DESIGN
C E H SCHLOSSER LEC 11/08/79 STUBBED
C E H SCHLOSSER LEMSCO 05/28/80 IMPLEMENTED
C
C
C METHOD
C -----
C
C CHECK DIAGNOSTIC COUNTERS AND PREPARE FOR NEXT PICTURE.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C NONE.
C
C
C EXTERNAL REFERENCES
C -----
C
C MONOTE
C MOCLRW
C
C
C EXCEPTIONS
C -----
C
C NONE.
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST 3 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C INCLUDE KOMOWW.LIST 3 COMMON OUTPUT WINDOW PACKETS
C INCLUDE WINDEF.LIST 3 DEFINE STRUCTURE OF WINDOW PACKETS
C INCLUDE NULCST.LIST 3 DEFINE NULL CHARACTER STRING
C VIA TO
C EXTERNAL PIC000. NULSUB
C
C
C
C PROCEDURE
C -----
C
C CALL TRACE
C
C
C ON RETURN, CALL PIC000 TO GET COMMANDS
C

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICP19
002

```
      CALL NVIATO(PIC000,NULSUB)
C
C
C ANY DIAGNOSTICS???
C
      IF(INDFATL.EQ.0) GO TO 920
      CALL MDNOTE( 'FATAL ERRORS -- NO PICTURE GENERATED')
      GO TO 990
920 IF(MDHARN.EQ.0) GO TO 960
      CALL MDNOTE( 'PREVIOUS WARNINGS -- NO PICTURE GENERATED')
      IF(MBATCH.EQ.0) WRITE(6,925)
925      FORMAT(4X,'**TRY AGAIN!')
      CALL MDCLR( NULCST)
      GO TO 990
C
C
C PRINT MSA OUTPUT WINDOW COORDINATES AND PREPARE FOR NEXT WINDOW
C
960 WRITE(6,965) MSAOWH(WLIN,WMIN)
965 FORMAT(1X,11X,'LINE ',14)
      WRITE(6,975) MSAOWH(WSAM,WMIN),MSAOWH(WSAM,WMAX)
975 FORMAT(1X,'SAMPLE ',14,9X,'SAMPLE ',14)
      WRITE(6,985) MSAOWH(WLIN,WMAX)
      WRITE(6,985)
985 FORMAT(1X)
      NWNDOW=NWNDOW+1
C
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
990 RETURN
      END
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICPRO
001

SUBROUTINE PICPRO(3 GENERATE PROFILES
U KOMD) 3 1: FIRST 3 CHARS OF COMMAND 0: SPACES

C
C
C
C
C

(E H SCHLOSSER)

INCLUDE KOMXQT.LIST
EXTERNAL PIC000.NULSUB
CALL TRACE

C
C

CALL NVIATO(PIC000.NULSUB)
CALL MDNOTE('PROFILE COMMAND NOT YET IMPLEMENTED')
RETURN
END

```
      SUBROUTINE PICPR3  & PROFILE NSS-DERIVED DATA (PHASE 3)
      -----
C
C (E H SCHLOSSER)
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
C -----
C
C      MDNOTE
C      NVIATO
C
C      INCLUDE KOMTBL.LIST
C      EXTERNAL PIC129.PICPR9
C      CALL TRACE
C
C
C
C      CALL MDNOTE('PROFILE.RADIANCE NOT YET IMPLEMENTED')
C      CALL NVIATO(PIC129.PICPR9)
C      RETURN
C      END
```

SUBROUTINE PICPR9 3 PROFILE HSS-DERIVED DATA (PHASE 9)

C
C
C (E H SCHLOSSER)
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
C -----
C
C MONOTE
C NVIATO
C
C
C EXTERNAL PIC000.NULSUB
C CALL TRACE
C
C
C
C CALL MONOTE('PROFILE NOT YET IMPLEMENTED')
C CALL NVIATO(PIC000.NULSUB)
C RETURN
C END


```
INCLUDE KONXQT.LIST      & COMMON PROGRAM EXECUTION COUNTERS. SWITCHES
INCLUDE KONNER.LIST     & COMMON ERTS SCENE PARAMETERS
INCLUDE KONKLS.LIST     & COMMON CLASSIFICATION INFO
INCLUDE KONTBL.LIST     & COMMON FACTOR TABLE
INCLUDE KOMIRT.LIST     & IRRADIANCE TRANSFORMATION COEFFICIENTS
INCLUDE NULCST.LIST     & DEFINE NULL CHARACTER STRING
```

C

C

C LOCAL DECLARATIONS

C

C

```
REAL ARAD      & ROTATION ANGLE IN RADIANs
REAL ADEG      & ROTATION ANGLE IN DEGREEs
```

C

C

C PROCEDURE

C

C

C

CALL TRACE

C

C

C CHECK IF FACTOR TABLE LOADED

C

C

```
IF((KLSSTYP.EQ.0).OR.      & NO CLASS TYPE FROM PREVIOUS FACTORING
& (KTBLTY.NE.'FACT')) & NO FACTOR TABLES LOADED
& CALL MDWARN('NO PREVIOUS FACTORING')
```

C

C

C GET FUNCTION TO MAXIMIZE UNDER ROTATION

C

```
KFUTYP='VAR'      & DEFAULT IS VARIMAX ROTATION
CALL GETSKH(KFUTYP,3,NULCST)
IF(KFUTYP.EQ.'DEG') GO TO 500      & NO MAXIMIZATION -- EXPLICIT ANGLE
IF(KFUTYP.NE.'VAR'.AND.KFUTYP.NE.'QUA') CALL WARN5(
& 'BAD MAXIMIZATION FUNCTION --')
```

C

C

C GET TYPE OF MATRIX USED IN EVALUATING FUNCTION

C

```
KMATYP='STR'      & DEFAULT IS FACTOR STRUCTURE
CALL GETSKH(KMATYP,3,NULCST)
IF(KMATYP.NE.'STR'.AND.KMATYP.NE.'COE') CALL WARN5(
& 'BAD EVALUATION MATRIX --')
```

C

C

C IS NORMALIZED MATRIX TO BE USED?

C

```
KNOTYP='NOR'      & DEFAULT IS NORMALIZED
CALL GETSKH(KNOTYP,3,NULCST)
IF(KNOTYP.NE.'NOR'.AND.KNOTYP.NE.'UNN'.AND.KNOTYP.NE.'RAW')
& CALL WARN5('BAD NORMALIZATION SPEC --')
```

C

C

C DRAIN SPECS FOR CURRENT COMMAND

```
C
      CALL GETSIN(ITEMP,+1,-1,'EXTRA ROTATE SPECIFICATION --')
C
C
C CHECK FOR DIAGNOSTICS
C
      IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE
      IF(INDTOTL.NE.0) GO TO 800
C
C
C BRANCH ON FUNCTION SPECIFICATION
C
      IF(KFUTYP.NE.'QUA') GO TO 300      & VARIMAX
C
C
C QUARTIMAX ROTATION
C
      IF(KMATYP.NE.'STR') GO TO 240      & COEFFICIENTS
C
C
C QUARTIMAX ROTATION OF FACTOR STRUCTURE
C
      IF(KNOTYP.NE.'NOR') GO TO 220      & UNNORMALIZED
C
C
C
      WRITE(6,215)
      215 FORMAT(' ROTATE. QUARTIMAX. STRUCTURE. NORMALIZED')
      CALL ROTCHX(FSTRUC,QUARTN,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
      CALL ROTCOL(FCNORM,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
      GO TO 700
C
C
      220 WRITE(6,225)
      225 FORMAT(' ROTATE. QUARTIMAX. STRUCTURE. UNNORMALIZED')
      CALL ROTCHX(FSTRUC,QUARTU,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
      CALL ROTCOL(FCNORM,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
      GO TO 700
C
C
C QUARTIMAX ROTATION OF FACTOR COEFFICIENTS
C
      240 IF(KNOTYP.NE.'NOR') GO TO 260      & UNNORMALIZED
C
C
C
      WRITE(6,245)
      245 FORMAT(' ROTATE. QUARTIMAX. COEFFICIENTS. NORMALIZED')
      CALL ROTCHX(FCNORM,QUARTN,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
      CALL ROTCOL(FSTRUC,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
      GO TO 700
C
C
      260 WRITE(6,265)
      265 FORMAT(' ROTATE. QUARTIMAX. COEFFICIENTS. UNNORMALIZED')
      CALL ROTCHX(FCNORM,QUARTU,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
      CALL ROTCOL(FSTRUC,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
      GO TO 700
```

```
C
C
C VARIMAX ROTATION
C 300 IF(KMATYP.NE.'STR') GO TO 340      8 COEFFICIENTS
C
C
C VARIMAX ROTATION OF FACTOR STRUCTURE
C      IF(KNOTYP.NE.'NOR') GO TO 320      8 UNNORMALIZED
C
C
C      WRITE(6,315)
315 FORMAT(' ROTATE, VARIMAX, STRUCTURE, NORMALIZED'//)
    CALL ROTCHX(FSTRUC,VARSON,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
    CALL ROTCOL(FCNORM,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
    GO TO 700
C
C
C 320 WRITE(6,325)
325 FORMAT(' ROTATE, VARIMAX, STRUCTURE, UNNORMALIZED'//)
    CALL ROTCHX(FSTRUC,VARSON,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
    CALL ROTCOL(FCNORM,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
    GO TO 700
C
C
C VARIMAX ROTATION OF FACTOR COEFFICIENTS
C 340 IF(KNOTYP.NE.'NOR') GO TO 380      8 UNNORMALIZED
C
C
C      WRITE(6,345)
345 FORMAT(' ROTATE, VARIMAX, COEFFICIENTS, NORMALIZED'//)
    CALL ROTCHX(FCNORM,VARSON,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
    CALL ROTCOL(FSTRUC,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
    GO TO 700
C
C
C 360 WRITE(6,365)
365 FORMAT(' ROTATE, VARIMAX, COEFFICIENTS, UNNORMALIZED'//)
    CALL ROTCHX(FCNORM,VARSON,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
    CALL ROTCOL(FSTRUC,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
    GO TO 700
C
C
C GET EXPLICIT ROTATION ANGLE(S)
C 500 CALL GETSSX(ADEOFM,1,..-360..+360..'BAD ROTATE ANGLE ---')
C
C
C GET OPTIONAL FINAL ROTATION ANGLE & INCREMENT
C
ADEO=ADEO=ADEOFM      8 DEFAULT FINAL ANGLE IS SAME AS INITIAL
CALL GETSSX(ADEO,1,..-360..+360..'BAD ROTATE FINAL ANGLE ---')
ADEOIN=1.             8 DEFAULT INCREMENT IS 1 DEGREE
CALL GETSSX(ADEOIN,1,..-360..+360..'BAD ROTATE INCREMENT ---')
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICROT
005

```
      WRITE(6,505) ADEOFM,ADEOTO,ADEOIN
505 FORMAT(' ROTATE, DEGREES',3(' ',F9.4))
C
C
C PERFORM EXPLICIT ROTATION(S) & EVALUATE FUNCTIONS
C
      IF(INDTOTL.NE.0) GO TO 600      & DIAGNOSTICS ENCOUNTERED!
      WRITE(6,515)
515 FORMAT(
& '0 DEGREES      '
& '      Q U A R T I M A X      V A R I M A X'
& 12X.' STRUCTURE COEFFICIENTS      STRUCTURE COEFFICIENTS'
& 12X.'NORM UNNORM NORM UNNORM      NORM UNNORM NORM UNNORM')
      MDEOFM=ADEOFM*1000.
      MDEOTO=ADEOTO*1000.
      MDEOIN=ADEOIN*1000.
      DO 540 MDEO=MDEOFM,MDEOTO,MDEOIN
      ADEO=FLOAT(MDEO)*.001
      ARAD=ADEO*3.14159265/180.
      CALL ROTCOL(FSTRUC,KCHACO,KFACCO,7,7,1,2,ARAD,FSTROT)
      CALL ROTCOL(FCNORM,KCHACO,KFACCO,7,7,1,2,ARAD,FCNROT)
      SUARTN=QUARTN(FSTROT,KCHACO,KFACCO,7,7,1,2)
      SUARTU=QUARTU(FSTROT,KCHACO,KFACCO,7,7,1,2)
      CUARTN=QUARTN(FCNROT,KCHACO,KFACCO,7,7,1,2)
      CUARTU=QUARTU(FCNROT,KCHACO,KFACCO,7,7,1,2)
      SARSON=VARSON(FSTROT,KCHACO,KFACCO,7,7,1,2)
      SARSQU=VARSQL(FSTROT,KCHACO,KFACCO,7,7,1,2)
      CARSON=VARSON(FCNROT,KCHACO,KFACCO,7,7,1,2)
      CARSQU=VARSQL(FCNROT,KCHACO,KFACCO,7,7,1,2)
      WRITE(6,525) ADEO.
& SUARTN,SUARTU,CUARTN,CUARTU,SARSON,SARSQU,CARSON,CARSQU
525 FORMAT(F9.3,1X,4F7.4,1X,4F7.4)
540 CONTINUE
      WRITE(6,555)
555 FORMAT(1X)
      GO TO 750
C
C
C PRINT ROTATION ANGLE IN DEGREES
C
      700 ADEO=ARAD*180./3.14159265
      WRITE(6,725) ADEO
      725 FORMAT(' (ROTATION ANGLE = ',F9.4,' DEGREES)')
C
C
C PRINT FACTOR STRUCTURE/COEFFICIENTS/MEANS
C
      750 CALL FACPRF(FSTROT,FCNROT,FCOROT,CMSTD,CMMEAN,KCHACO,KFACCO,7,7)
C
C
C ASSIGN FACTOR COEFFICIENTS TO LINEAR TRANSF WEIGHTS & UPDATE WEIGHTED GAINS
C
      DO 770 NCH=1,NERCHA
      DO 780 NLF=1,2
      RTLWOT(NCH,NLF)=FCOROT(NCH,NLF)
      780 LRTWIZ(NCH,NLF)=RTLWOT(NCH,NLF)*RTLOAN(NLF)*2.**12
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICROT
008

```
770 CONTINUE
    GO TO 800
C
C
C CHECK DIAGNOSTIC COUNTERS
C
800 IF(NDWARN.EQ.0) GO TO 820
    CALL MDNOTE('PREVIOUS WARNINGS -- NO ROTATION')
    IF(NDATCH.EQ.0) WRITE(6,815)
815 FORMAT(' ...TRY AGAIN')
    GO TO 890
820 IF(NDFATL.EQ.0) GO TO 890
    CALL MDNOTE('PREVIOUS FATAL ERRORS -- NO ROTATION')
    GO TO 890
890 IF(MCHECK.EQ.0) GO TO 890
    CALL MDNOTE('CHECKOUT MODE -- NO ROTATION')
C
C
C CLEAR WARNINGS
C
890 CALL MDCLR(MNULCST)
C
C
C RETURN
C
900 KOND= '
    RETURN
    END
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICTOT
001

```
      SUBROUTINE PICTOT( 8 TOTAL TABULATIONS  
U KOND)      8 1: FIRST 3 CHARS OF COMMAND  0: SPACES  
-----  
C  
C  
C (E H SCHLOSSER)  
C  
C  
      INCLUDE KONXQT.LIST  
      EXTERNAL PIC000.NULSUB  
      CALL TRACE  
C  
C  
      CALL NVIATO(PIC000.NULSUB)  
      CALL HDNOTE('TOTAL COMMAND NOT YET IMPLEMENTED')  
      KOND= '  
      RETURN  
      END
```


DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICTOS
001

```
      SUBROUTINE PICTOS  & TOTAL TABULATIONS  
-----  
C  
C  
C (E H SCHLOSSER)  
C  
C  
      INCLUDE KONXGT.LIST  
      EXTERNAL PIC000.NULSUB  
      CALL TRACE  
C  
C  
      CALL NVIATO(PIC000.NULSUB)  
      CALL MDNOTE('TOTAL COMMAND NOT YET IMPLEMENTED')  
      RETURN  
      END
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PICX01
002

```
C PROCEDURE
C -----
C
C
C IDENTIFY PROGRAM
C
C     CALL PSTART( 'DAM PICTAB(0009)')
C
C
C ON RETURN, CALL PIC000 TO GET DEFAULT/USER COMMANDS
C
C     CALL NVIATO( PIC000.NULSUB)
C
C
C OPEN FILES & LOAD REGISTRATION PARAMETERS
C
C     CALL OPEN3
C     CALL LOREGN
C     IF(MDATAC.NE.0) GO TO 300      & DATA/CHECKOUT MODE
C
C
C IDENTIFY ERTS SCENE
C
C     WRITE(6,225)
C     225 FORMAT(1X)      & SKIP LINE
C     CALL IOLU3( 6)
C     CALL IDERTS( 6)
C
C
C
C QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE
C
C     300 CALL SYSADD(LOCFIL, 'MACDAM','DEF-PICTAB',. )
C         IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL, 'DAM','DEF-PICTAB',. )
C         IF(LOCFIL.LE.0) CALL MOFATL( 'NO DEFAULT COMMANDS')
C
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
C     RETURN
C     END
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

10CPIC
002

```
INCLUDE KOMXQT.LIST      3 COMMON PROGRAM SWITCHES.COUNTERS
INCLUDE KOMNER.LIST      3 ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST      3 CLASSIFICATION INFORMATION
INCLUDE KOMIRT.LIST      3 IRRADIANCE TRANSFORM COEFFICIENTS
INCLUDE KOMSYM.LIST      3 COMMON SYMBOL TABLE
INCLUDE WINDEF.LIST      3 DEFINE TWO-DIMENSIONAL WINDOW
INCLUDE KOMOHV.LIST      3 DEFINE OUTPUT WINDOW PACKET
INCLUDE ICBUF1.LIST      3 DECLARE CHARACTER BUFFER

C
C
C LOCAL DECLARATIONS
C -----
C
  INTEGER NAMFIL (2)      3 ALTERNATE PRINT FILE NAME
  INTEGER NRWCHN          3 NUMBER OF RAW CHANNEL
  INTEGER NLCHAN          3 NUMBER OF LINEAR CHANNEL
  INTEGER NPCHAN          3 NUMBER OF POLAR CHANNEL
  INTEGER NLC             3 NUMBER OF LIMIT CHANNEL
  INTEGER NAXIS          3 AXIS NUMBER
  CALL TRACE

C
C
C PROCEDURE
C -----
C
C INITIALIZE FILE NAME AND CHARACTER BUFFER
C
  CALL CST4IN (NAMFIL.(1).(2), UNIT.1)
  CALL CBINIT (ICBUF1)

C
C CHECK FOR VALID OUTPUT UNIT/PRINT FILE
C
  IF (UNIT.EQ.6.OR.
    * UNIT.GE.10.AND.UNIT.LE.(10+HALTH-1)) GO TO 200
    CALL MDATL (CST4IN(UNIT.(1)), ' ' IS BAD UNIT IN 10CPIC')
    GO TO 900

C
C
C OUTPUT SHARPENING SPECS. IF PRESENT
C
200 CALL CB4CST (ICBUF1, ' ')
   DO 250 NRWCHN=1,NRCHA
     IF (IRSF12(NRWCHN.1).EQ.0.AND.
       * IRSF12(NRWCHN.2).EQ.0) GO TO 250 3 NO COEFFS FOR CHAN
       IF (LENCST(ICBUF1.2).GT.1) CALL CB4CST (ICBUF1,
         * .....')
       CALL CB4CST (ICBUF1, 'SHA.')
       CALL CB4IN (ICBUF1, NRWCHN.1)
       CALL CB4CST (ICBUF1, 'SAM.')
       CALL CB4RL (ICBUF1,
         * FLOAT(IRSF12(NRWCHN.1))*2.**-12.1.4)
       CALL CB4CST (ICBUF1, ' ')
       CALL CB4RL (ICBUF1,
         * FLOAT(IRSF12(NRWCHN.2))*2.**-12.1.4)
```

```

290      CONTINUE
        CALL CB4CST (ICBUF1,  '1')
        IF (LENCST(ICBUF1,3).LE.2) GO TO 290      & BLANK BUFFER
        IF (IUNIT.EQ.6) CALL ERPRNT (1.22,ICBUF1)
        IF (IUNIT.NE.6) CALL ERPRTA (NAMFIL,1.22,ICBUF1)

C
C
C OUTPUT TRANSFORMATION SPECS
C ---SPECS FOR LINEAR TRANSFORMATIONS
C
290 IF (IRTTY.EQ.'RAW') GO TO 400      & NO TRANSFORMATIONS
      DO 320 NLCHAN=1,2
        CALL CBINIT (ICBUF1)
        CALL CB4CST (ICBUF1,  '(LIN,')
        CALL CB4IN (ICBUF1,  NLCHAN,1)
        CALL CB4CST (ICBUF1,  '.WEI,')
        DO 300 NRCHN=1,NERCHA
          CALL CB4RL (ICBUF1,  RTLWGT(NRCHN,NLCHAN),1.4)
          CALL CB4CST (ICBUF1,  ',')
300      CONTINUE
        CALL CB4CST (ICBUF1,  'GAIN,')
        CALL CB4RL (ICBUF1,  RTLGAN(NLCHAN),1.3)
        CALL CB4CST (ICBUF1,  '.BIAS,')
        CALL CB4RL (ICBUF1,
                   FLOAT(LRTB12(NLCHAN))*2.**-12,1.3)
        CALL CB4CST (ICBUF1,  ',')
        IF (IUNIT.EQ.6) CALL ERPRNT (1.22,ICBUF1)
        IF (IUNIT.NE.6) CALL ERPRTA (NAMFIL,1.22,ICBUF1)
320      CONTINUE

C
C ---SPECS FOR POLAR TRANSFORMATIONS
C
      IF (IRTTY.NE.'POL') GO TO 400      & NO POLAR TRANSFORMATION
      CALL CBINIT (ICBUF1)
      CALL CB4CST (ICBUF1,  '1')
      DO 350 NPCHAN=1,2
        IF (NPCHAN.EQ.2) CALL CB4CST (ICBUF1,  '....')
        CALL CB4CST (ICBUF1,  'POL,')
        CALL CB4IN (ICBUF1,  NPCHAN,1)
        CALL CB4CST (ICBUF1,  '.GAIN,')
        CALL CB4RL (ICBUF1,
                   FLOAT(NRT012(NPCHAN))*2.**-12,1.3)
        CALL CB4CST (ICBUF1,  '.BIAS,')
        CALL CB4RL (ICBUF1,
                   FLOAT(NRT024(NPCHAN))*2.**-24,1.3)
350      CONTINUE
        CALL CB4CST (ICBUF1,  '1')
        IF (IUNIT.EQ.6) CALL ERPRNT (1.22,ICBUF1)
        IF (IUNIT.NE.6) CALL ERPRTA (NAMFIL,1.22,ICBUF1)

C
C
C IDENTIFY CHANNEL TYPE AND NUMBER(S)
C
400 CALL CBINIT (ICBUF1)
      CALL CB4CST (ICBUF1,  '(CHAN,')
      CALL CB4CST (ICBUF1,  IRTTY,(1),(3))

```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

10CPIC
004

```
DO 450 NLC=1,NLIMCH  
  CALL CB4CST (ICBUF1, '...')  
  CALL CB4IN (ICBUF1, LINCH(NLC),1)  
450 CONTINUE
```

C

C

C IDENTIFY RADIANCE LIMIT VALUES AND SYMBOLS FOR FIRST CHANNEL

C

```
CALL CB4CST (ICBUF1, '...RAD.')  
CALL CB4IN (ICBUF1, LCVLO1,1)  
CALL CB4CST (ICBUF1, '...')  
CALL CB4CST (ICBUF1, KSYM(LCVLO1+1),(1),(1))  
CALL CB4CST (ICBUF1, '...')  
CALL CB4IN (ICBUF1, LCVHI1,1)  
CALL CB4CST (ICBUF1, '...')  
CALL CB4CST (ICBUF1, KSYM(LCVHI1+1),(1),1)
```

C

C

C IDENTIFY RADIANCE LIMIT VALUES FOR ANY OTHER CHANNELS

C

```
IF (NLIMCH.LT.2) GO TO 600  
DO 550 NLC=2,NLIMCH
```

```
C S CALL XREG77(2) 2=2 *** DUMP REG X2 *** 2=2  
C S CALL CB4CST (ICBUF1, '...')  
CALL CB4CST (ICBUF1, '...1,1')  
C S CALL XREG77(2) 2=2 *** DUMP REG X2 *** 2=2  
CALL CB4IN (ICBUF1, LCVLO(NLC),1)  
C S CALL XREG77(2) 2=2 *** DUMP REG X2 *** 2=2  
C S CALL CB4CST (ICBUF1, '...')  
CALL CB4CST (ICBUF1, '...1,1')  
CC S CALL XREG77(2) 2=2 *** DUMP REG X2 *** 2=2  
CALL CB4IN (ICBUF1, LCVHI(NLC),1)  
C S CALL XREG77(2) 2=2 *** DUMP REG X2 *** 2=2  
550 CONTINUE
```

C

C

C IDENTIFY SPACING

C

```
600 CALL CB4CST (ICBUF1, '...SPA')  
DO 650 NAXIS=1,2  
  CALL CB4CST (ICBUF1, '...')  
  SPA=FLOAT(MSAQHW(NAXIS,WSP100))/100.  
  IF (SPA.EQ.AINT(SPA)) CALL CB4IN (ICBUF1, IFIX(SPA),1)  
  IF (SPA.NE.AINT(SPA)) CALL CB4RL (ICBUF1, SPA,1,2)  
650 CONTINUE
```

C

C

C IDENTIFY ORIGIN

C

```
CALL CB4CST (ICBUF1, '...ORIG.SCAN')  
CALL CB4IN (ICBUF1, MSAQHW(WLIN,WORIG),1)  
CALL CB4CST (ICBUF1, '...')  
CALL CB4IN (ICBUF1, MSAQHW(WSAM,WORIG),1)  
CALL CB4CST (ICBUF1, '...')
```

C

C

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

10CP1C
009

C OUTPUT CHANNEL/RADIANCE/SPACING/ORIGIN INFO

C

IF (IUNIT.EQ.6) CALL ERPRNT (1.22.ICBUF1)

IF (IUNIT.NE.6) CALL ERPRTA (NAMFIL.1.22.ICBUF1)

C

C

C NORMAL RETURN

C

900 RETURN

END

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

OPRPIC
002

```
C -----  
C  
C      CALL TRACE  
C  
C  
C NO MORE THAN MALTHI ALTERNATE PRINT FILES FOR PICTAB  
C  
C      MALTH=MAX0(MALTH,1)  
C      MALTH=MIN0(MALTH,MALTHI)  
C  
C  
C NO MORE THAN KPAGHI COLUMNS PER PAGE FOR PICTAB  
C  
C      KPAGE=MAX0(KPAGE,1)  
C      KPAGE=MIN0(KPAGE,KPAGHI)  
C  
C  
C OPEN FILE(S)  
C  
C      IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE  
C      CALL OPENPR  
C  
C  
C IDENTIFY ERTS SCENE  
C  
C      WRITE(10,225)  
C      225 FORMAT(1X)      & SKIP LINE  
C      CALL IDLU3(10)  
C      CALL IDERTS(10)  
C  
C  
C 900 RETURN  
C      END
```

C PROGRAM CONTROL
C -----
C

C HISTORY
C -----
C

C	E H SCHLOSSER	LEC	09/27/73	ORIGINAL CODE IN COEF & CTROL
C	E H SCHLOSSER	LEC	09/30/79	COEF & CTROL COMBINED IN CONTROL
C	E H SCHLOSSER	LEC	11/14/79	UPGRADE DOCUMENTATION

C METHOD
C -----
C

C THIS PROGRAM ADJUSTS AND/OR DIAGRAMS A NETWORK OF CONTROL POINTS FOR A
C LANDSAT MSS SCENE. THE REGISTRATION PARAMETERS FROM THE ADJUSTMENT ARE
C STORED ON UNIT 8 (TEMPORARY FILE) FOR USE IN THE SAME RUN BY OTHER
C PROGRAMS OF THE DAH PACKAGE.

C THIS PROGRAM IS LIMITED TO THAT PART OF THE WORLD COVERED BY THE
C CLARKE 1866 SPHEROID (NORTH AMERICA).

C UP TO 350 POINTS MAY BE USED. POINTS WITH USER-ASSIGNED NUMBERS
C BETWEEN +1 AND +999 ARE CONTROL POINTS AND WILL BE USED IN THE
C ADJUSTMENT. POINTS WITH NUMBERS BETWEEN -1 AND -999 ARE CHECK POINTS
C AND WILL NOT BE USED IN THE ADJUSTMENT. RESIDUAL ERRORS
C FOR CONTROL POINTS AND CHECK POINTS ARE COMPUTED SEPARATELY.
C AT LEAST 5 CONTROL POINTS MUST BE USED. IF ONE NETWORK IS TO
C BE USED FOR ALL 4 STRIPS OF A SCENE IT SHOULD CONTAIN AT LEAST 8
C CONTROL POINTS, WITH AT LEAST 1 CONTROL POINT IN EACH OF THE 4
C CCT STRIPS.

C ERTS CONVENTIONS FOR ATTITUDE AND HEADING ARE AS FOLLOWS:
C POSITIVE PITCH IS NOSE DOWN
C POSITIVE ROLL IS CLOCKWISE VIEWED FROM BEHIND
C POSITIVE YAW IS COUNTERCLOCKWISE VIEWED FROM ABOVE
C POSITIVE HEADING IS CLOCKWISE VIEWED FROM ABOVE

C IF NERTS(3) IS NEGATIVE BUT OTHERWISE VALID, NON-LINEAR CORRECTIONS ARE NOT
C MADE BEFORE PERFORMING THE ADJUSTMENT. THE RESULTS OF SUCH AN ADJUSTMENT
C MAY NOT BE USED WITH DAH.CLASSIFY.

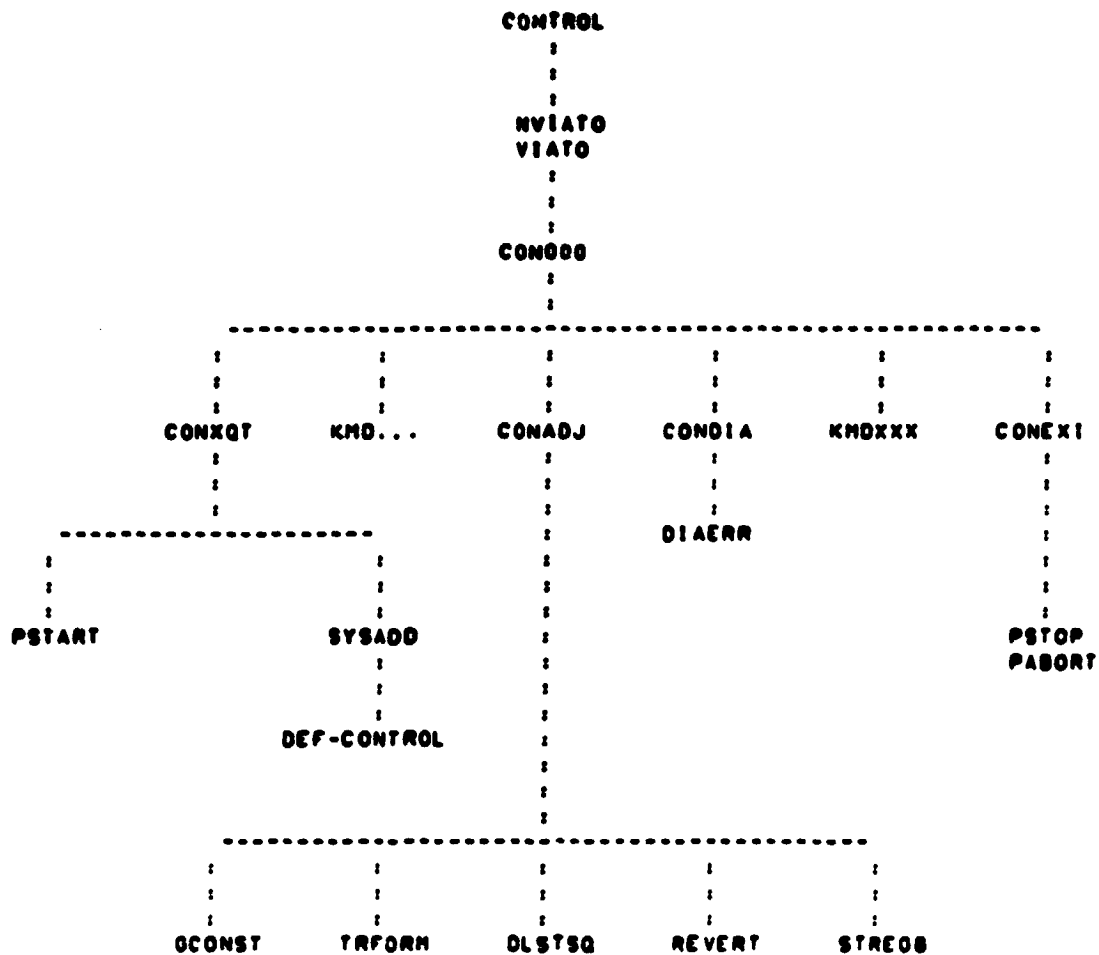
C MACHINE-DEPENDENT CODE
C -----
C

C NONE.
C

C EXTERNAL REFERENCES
C -----
C

C	NVIATO	3	NAME	'VIA'	'TO'	ROUTINES
C	VIATO	3	CALL	'VIA'	'TO'	ROUTINES
C			VIA		TO	

CONTROL HIERARCHY



PROGRAM CONTROL/VIRTUAL

HISTORY

E H SCHLOSSER LCC 00/02/74 ORIGINAL CODE
E H SCHLOSSER LEC 11/00/70 SNAP.FZINI: NO 'N' IN DEMAND

METHOD

CONSTRUCT SNAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT SXQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE SNAP & SXQT COMMANDS TO TEMPORARY FILE 20.
SADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 8-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS & FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IONS & INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS & TERMINATE PROGRAM EXECUTION
DAM.CONTROL-MAP & SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAPOPT & STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS BASG-D & SPREP-D.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

1 2 = REAL TIME
1 3 = LOW EXEC
1 4 = DEMAND
1 5 = DEADLINE BATCH
1 6 = BATCH

(SXQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER A5 IN

(MASTER BIT NOTATION.

LOCAL DECLARATIONS

```

      AXRS
S(00) . D-BANK
SSSH FORM 0.0.0.10
      111111222222333333444444555555666666777777888888999999
LABSDF SSSH 090.1.'F'.0 . LABEL. 1 WD. FORTRAN. FIELDATA
LABING 'SDF'
MAPSDF SSSH 000.0.0.0 . DATA. 9 WDS. . FIELDATA
MAPING 'XQTS: MAP.FZN DAM.CONTROL-MAP.CONTROL . :XQTS'
ADDSD SSSH 000.0.0.0
ADDING 'XQTS: ADD DAM.SYS-MAPOPT . :XQTS'
XQTSDF SSSH 000.0.0.0
XQTING 'XQTS: XQT.1 CONTROL . :XQTS'
E0FSDF - 0 . END-OF-FILE STOP WORD
PF FORM 12.0.10
CSFASO 'BASO.T 20. . .
CSFADD 'SADD 20. . .
SAVREG RES 1
IOPKT ISOD '20'.MS 33.LABSDF.'0' 0
  
```

PROCEDURE

```

S(01) . I-BANK
CONTROL LA.U . AG 10 .
      TNE.U . A4.4 . SKIP NEXT INST IF A4<14 (NOT DEMAND)
      SA.S2 . AG.MAPING*2 . DEMAND! BLANK OUT N OPTION
      LA . AG.(CSFASO) . ADDRESS OF BASO IMAGE
      ER . CSFS . DO IT
      SA . AG.SAVREG . STORE S
      PPRINT (PF 2.1.SAVREG) . PRINT BASO STATUS

      OETOPT . LOAD OPT LTRS INTO A2.A3.A4

PUTOPT DS . A2.XQTING*2 . STORE OPTION LETTERS INTO XQTS IMAGE
      SA . A4.XQTING*4 . (3 WORDS -- MAX 18 OPT LETTERS)

WRITE LA . AG.(IOPKT) . ADDRESS OF I/O PACKET
      ER . IOWS . WRITE SDF IMAGES TO 20.

ADD LA . AG.(CSFADD) . ADDRESS OF SADD IMAGE
      ER . CSFS . DO IT
      ER . EXITS

      END CONTROL
  
```

CONTROL OVERLAY STRUCTURE

HISTORY

E H SCHLOSSER	LEC	03/19/75	ORIGINAL CODE
E H SCHLOSSER	LEC	07/14/78	CHANGE OVERLAYS TO REDUCE THRASHING
E H SCHLOSSER	LEC	01/31/79	MACRO COMMANDS & TIME COMMAND
J C CRISP	LEC	11/29/79	PEEK/POKE/GEOMETRY/CENTER/SIZE/IF/FI

LIB DAM.

SEG S-MAIN

IN DAM.CONTROL/ . MAIN PROGRAM
 IN DAM.NVIATO . NAME/CALL 'VIA' AND 'TO' SUBROUTINES
 IN DAM.NULSUB . DO NOTHING
 IN DAM.SYS-BLOCK . BLOCK DATA SUBROUTINE

. MONITOR FOR PHASE 0.1.2.9 COMMANDS -----

IN DAM.CON000 . CALL USER-SPECIFIED PHASE 0 ROUTINE
 IN DAM.NTABS/DAM . DAM UNIT # TABLE GOES IN SAME SEG W/ FORTRAN I/O

. PHASE 0.1.2.9 COMMANDS (FORTRAN I/O ALLOWED) -----

SEG S-XQTEXT1

IN DAM.CONXQT . CONTROL INITIALIZATION ROUTINE
 IN DAM.CONEXI . CONTROL TERMINATION ROUTINE

SEG S-KMD---S-XQTEXT1

IN DAM.KMDATT . GET/CHECK ATTITUDE (PITCH & ROLL)
 IN DAM.KMDCEN . GET/CHECK CENTER SCAN COORDINATES
 IN DAM.KMDCLE . CLEAR WARNINGS/ERRORS
 IN DAM.KMDEXP . EXPLAIN PROGRAM/COMMAND
 IN DAM.KMDFI . END IF...FI BLOCK
 IN DAM.KMDGEO . GET/CHECK GEOMETRY
 IN DAM.KMDIF . BEGIN IF...FI BLOCK
 IN DAM.KMDNEW . PRINT NEWS
 IN DAM.KMDNEX . CONDITIONALLY PERFORM NEXT COMMAND
 IN DAM.KMDOFF . TURN OFF MODE SWITCH(ES)
 IN DAM.KMDON . TURN ON MODE SWITCH(ES)
 IN DAM.KMDPEE . PEEK AT LABELLED COMMON
 IN DAM.KMDPOI . GET CONTROL/CHECK/QUERY POINT
 IN DAM.KMDPOK . POKE AT LABELLED COMMON
 IN DAM.KMDSCE . GET/CHECK SCENE NUMBER & NUMBER OF SAMPLES PER SCENE
 IN DAM.KMDSIZ . GET/CHECK SIZE IN SCAN COORDINATES
 IN DAM.KMDTIM . PRINT CLOCK TIME & CHARGE TIME
 IN DAM.KMDXXX . MACRO COMMANDS
 IN DAM.KMDZON . GET/CHECK UTM PROJECTION ZONE
 IN DAM.KMDOAO . DYNAMIC 3ADD

**DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**CONTROL -MAP
002**

**SEG S-CONADJ*.S-XQTEXI
IN DAM.CONADJ . ADJUST NETWORK**

**SEG S-CONDIA*.S-XQTEXI
IN DAM.CONDIA . DIAGRAM NETWORK/ERRORS**

**DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**CONTROL -MAP/VIRTUAL
001**

IN DAM.CONTROL/VIRTUAL


```

      INTEGER LSSTAT      & READS STATUS ('EOF' MEANS END-OF-FILE)
      INTEGER KASE        & MODIFIED I-C-E OF FIRST CHAR OF COMMAND

C
C
C PROCEDURE
C -----
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
C   CALL TRACE
C   CALL NANSUB      & CALL TO NULSUB DOES NOTHING
C
C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C
C   KOMD=' NUL'      & IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
C   CALL READ5(LSSTAT,NULCST)  & FILL UNIT 5 BUFFER, NO CUE MESSAGE
C   IF(LSSTAT.NE.' ') KOMD='EXI'
C   -IF(KOMD.NE.'EXI') CALL GET5AL(KOMD.(13), NULCST)  & GET 3 ALPHA CHARS
C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C
C   KASE=ICE(KOMD)-ICE('A')+1      & A TO Z = 1 TO 26
C
C CASE STATEMENT ON MODIFIED ICE OF COMMAND'S FIRST CHARACTER
C
C   IF((KASE.LT.1).OR.(KASE.GT.26)) KASE=27      & NOT ALPHA
C   GO TO(
C     0 401,402,403,404,405,406,407,408,409,410,
C     1 411,412,413,414,415,416,417,418,419,420,
C     2 421,422,423,424,425,426,427)
C   & .KASE
C
C DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOMD TO BLANK
C
C 401 CONTINUE &*** A
C   IF(KOMD.EQ.'ADJ') CALL CONADJ(KOMD)      & ADJUST
C   IF(KOMD.EQ.'ATT') CALL XNDATT(KOMD)     & ATTITUDE
C   GO TO 800
C
C 402 CONTINUE &*** B
C   GO TO 800
C
C 403 CONTINUE &*** C
C   IF(KOMD.EQ.'CEN') CALL KHCEN(KOMD)      & CENTER
C   IF(KOMD.EQ.'CLE') CALL KHCLE(KOMD)     & CLEAR
C   GO TO 800
C
C 404 CONTINUE &*** D
C   IF(KOMD.EQ.'DIA') CALL CONDIA(KOMD)    & DIAGRAM
C   GO TO 800
C

```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CON000
003

```
405 CONTINUE 3*** E
   IF(KOMD.EQ.'EXI') CALL CONEXI(KOMD)      3 EXIT
   IF(KOMD.EQ.'EXP') CALL KMDEXP(KOMD)     3 EXPLAIN
   GO TO 800
C
406 CONTINUE 3*** F
   IF(KOMD.EQ.'FI ') CALL KMDFI (KOMD)     3 FI (ENDIF)
   GO TO 800
C
407 CONTINUE 3*** G
   IF(KOMD.EQ.'GEO') CALL KMDOGEO(KOMD)    3 GEOMETRY
   GO TO 800
C
408 CONTINUE 3*** H
   GO TO 800
C
409 CONTINUE 3*** I
   IF(KOMD.EQ.'IF ') CALL KMDIF (KOMD)     3 IF
   GO TO 800
C
410 CONTINUE 3*** J
411 CONTINUE 3*** K
412 CONTINUE 3*** L
413 CONTINUE 3*** M
   GO TO 800
C
414 CONTINUE 3*** N
   IF(KOMD.EQ.'NEW') CALL KMDNEW(KOMD)     3 NEWS
   IF(KOMD.EQ.'NEX') CALL KMDNEX(KOMD)    3 NEXT
   GO TO 800
C
415 CONTINUE 3*** O
   IF(KOMD.EQ.'OFF') CALL KMDOFF(KOMD)    3 OFF
   IF(KOMD.EQ.'ON ') CALL KMDON (KOMD)    3 ON
   GO TO 800
C
416 CONTINUE 3*** P
   IF(KOMD.EQ.'PEE') CALL KMDPPEE(KOMD)    3 PEEK
   IF(KOMD.EQ.'POI') CALL KMDDPOI(KOMD)   3 POINT
   IF(KOMD.EQ.'PO ') CALL KMDDPOI(KOMD)   3 POINT
   IF(KOMD.EQ.'P ') CALL KMDDPOI(KOMD)   3 POINT
   IF(KOMD.EQ.'POK') CALL KMDDPOK(KOMD)   3 POKE
   GO TO 800
C
417 CONTINUE 3*** Q
418 CONTINUE 3*** R
   GO TO 800
C
419 CONTINUE 3*** S
   IF(KOMD.EQ.'SCE') CALL KMDSCE(KOMD)    3 SCENE
   IF(KOMD.EQ.'SIZ') CALL KMDSIZ(KOMD)    3 SIZE
   GO TO 800
C
420 CONTINUE 3*** T
   IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD)    3 TIME
   GO TO 800
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CON000
004

```
C
421 CONTINUE 8000 U
422 CONTINUE 8000 V
423 CONTINUE 8000 W
424 CONTINUE 8000 X
425 CONTINUE 8000 Y
    GO TO 800

C
426 CONTINUE 8000 Z
    IF(KOMD.EQ.'ZON') CALL KMDZON(KOMD)      @ ZONE
    GO TO 800

C
427 CONTINUE 8000 NOT ALPHABETIC
    IF(KOMD.EQ.'SAD') CALL KMD0AD(KOMD)     @ SADD
    IF(KOMD.NE.' ') CALL KMDPOL(KOMD)      @ POINT

C
C
C IF COMMAND WAS NOT FOUND. TRY MACRO-COMMAND
C
800 IF(KOMD.NE.' ') KOMD='CON-'           @ 1ST 3 CHARS OF PROG NAME PLUS '--'
    IF(KOMD.NE.' ') CALL KMDXXX(KOMD)      @ MACRO COMMAND HANDLER

C
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
    IF(KOMD.NE.' ') CALL WARN5('INVALID COMMAND --')

C
C
C RETURN TO MAIN FOR CALL VIA/IO NAMED SUBROUTINE IN ANY OVERLAY
C
    RETURN
    END
```



```
C -----
C
C 1. INVALID VALUES FOR ANY OF THE FOLLOWING GENERATE 'WARNING' DIAGNOSTICS:
C     SCENE NUMBER
C     SAMPLES PER SCENE
C     PITCH
C     ROLL
C     NUMBER OF CONTROL POINTS
C     GEOMETRY
C
C 2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.
C
C GLOBAL DECLARATIONS
C -----
C
C     INCLUDE KOMXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C     INCLUDE KOMNER.LIST     & COMMON ERTS SCENE PARAMETERS
C     INCLUDE KOMFIT.LIST     & COMMON ADJUSTMENT/REGISTRATION PARAMETERS
C     INCLUDE KOMNET.LIST     & COMMON CONTROL NETWORK COORDINATE TABLE
C     INCLUDE WINDEF.LIST     & DEFINE STRUCTURE OF WINDOW PACKETS
C     INCLUDE TRFORM.LIST     & DEFINE COORDINATE TRANSFORMATION FUNCTIONS
C     INCLUDE NULCST.LIST     & DEFINE NULL CHARACTER STRING
C
C LOCAL DECLARATIONS
C -----
C
C     INTEGER NDSAVE          & TEMPORARY SAVE FOR CONTENTS OF NDTOTL ON ENTRY
C
C PROCEDURE
C -----
C
C     CALL TRACE
C
C CHECK FOR VALID SCENE AND GEOMETRY
C
C     NDSAVE=NDTOTL
C     IF((NERTS(1).LT.1).OR.
C     & (NERTS(1).GT.9)) CALL MDWARN('SCENE NOT DEFINED')
C     IF((NERGEO.NE.'ERT').AND.
C     & (NERGEO.NE.'HOM').AND.
C     & (NERGEO.NE.'LCC').AND.
C     & (NERGEO.NE.'PS ').AND.
C     & (NERGEO.NE.'SOM').AND.
C     & (NERGEO.NE.'UTM')) NERGEO='BAD'
C     IF(NERGEO.EQ.'BAD'.AND.
C     & (NERSAM.LT.3300)) NERGEO='ERT' & TEMP PATCH FOR OLD BIP FORMAT SCENES
C     IF(NERGEO.EQ.'BAD') CALL MDWARN('GEOMETRY NOT DEFINED')
C
C CHECK 'ERT' ATTITUDE AND SIZE
C
C     IF(NERGEO.NE.'ERT') GO TO 150
```

```

    IF((ABS(PITDEG).GT.9.).OR.(ABS(ROLDEG).GT.9)) CALL MDWARN(
    - 'ATTITUDE NOT SPECIFIED WITH ERT GEOMETRY')
    IF((NERSAM.LT.100).OR.(NERSAM.GT.10000)) CALL MDWARN(
    - 'SIZE (SAMPLES) NOT DEFINED')
    GO TO 190
  
```

C
 C
 C

C CHECK NON-'ERT' ATTITUDE, SIZE, AND CENTER

```

150 IF((ABS(PITDEG).LT.9.).AND.(ABS(ROLDEG).LT.9.)) GO TO 160
    PITDEG=0.
    ROLDEG=0.
160 IF(ABS(PITDEG)+ABS(ROLDEG).NE.0.) CALL MDWARN(
    - 'NON-ZERO ATTITUDE WITH' . ' CBS4CS(NERGEO.(1),(3))' ' GEOMETRY')
    IF((NERLIN.LT.100).OR.(NERLIN.GT.10000)) CALL MDWARN(
    - 'SIZE (LINES) NOT DEFINED')
    IF((NERSAM.LT.100).OR.(NERSAM.GT.10000)) CALL MDWARN(
    - 'SIZE (SAMPLES) NOT DEFINED')
    IF((CTRLIN.LT.50.).OR.(CTRLIN.GT.5000.)) CALL MDWARN(
    - 'CENTER (LINE) NOT DEFINED')
    IF((CTRSAM.LT.50.).OR.(CTRSAM.GT.5000.)) CALL MDWARN(
    - 'CENTER (SAMPLE) NOT DEFINED')
  
```

C
 C
 C

C CHECK NUMBER OF CONTROL POINTS

```

190 IF(NCTLPT.LT.5) CALL MDWARN( 'NOT ENOUGH CONTROL POINTS')
    IF((NDSAVE.NE.NDTOTL).AND.
    & (NDBATCH.NE.0)) CALL MDWATL( 'INCOMPLETE NETWORK')
  
```

C
 C
 C

C DETERMINE WHICH POINTS TO PRINT RESIDUAL ERRORS FOR

```

    NODT='ALL'
    CALL GETSKH(NODT,(3), NULCST)
    IF(NODT.EQ.'CON') GO TO 310      & JUST CONTROL POINTS
    IF(NODT.EQ.'CHE') GO TO 320      & JUST CHECK POINTS
    IF(NODT.EQ.'ALL') GO TO 330      & ALL POINTS
    IF(NODT.EQ.'NON') GO TO 340      & NONE
    CALL WARN5( 'BAD ADJUST SPECIFICATION --')
    GO TO 900
310 NPRPOS=1
    NPRNEQ=0
    IF(NCTLPT.GT.0) GO TO 400
    CALL MDWARN( 'NO CONTROL POINTS')
    GO TO 900
320 NPRNEQ=1
    NPRPOS=0
    IF((NETHI-NCTLPT).GT.0) GO TO 400
    CALL MDWARN( 'NO CHECK POINTS')
    GO TO 900
330 NPRPOS=1
    NPRNEQ=1
    GO TO 400
340 NPRPOS=0
    NPRNEQ=0
  
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CONADJ
884

```
C
C
C INITIALIZE
C
400 IF(INDSAVE.NE.NOTOTL) GO TO 900
   NHNDOW=ABS(NHNDOW)
   NERSAT(1)=.
   NERSAT(2)=.
   ALTKM=0.
   ALTSAM=0.
   CALL OCONST
   ROLRAD=ROLDG0*.0174532921
   NHNDOW=ABS(NHNDOW)
   CALL HOUNIT( 0.8)
   WRITE(6,415) NERTS
415 FORMAT(' SCENE ',11,J4,'-',J5)
C
C ESTIMATE CONTROL NETWORK CENTROID, COVERAGE, AND STM CENTRAL MERIDIAN
C
   CTDLIN=0.
   CTDSAM=0.
   CTDLAT=0.
   CTDLON=0.
   DO 610 K=1,NETM1
   IF(NETPT(K).LE.0) GO TO 610      & CHECK POINT
   CTDLIN=CTDLIN+ADJNET(WLIN,K)
   CTDSAM=CTDSAM+ADJNET(WSAM,K)
   CTDLAT=CTDLAT+OEDNET(WLAT,K)
   CTDLON=CTDLON+OEDNET(WLON,K)
610 CONTINUE
   CTDLIN=CTDLIN/FLOAT(NCTLPT)
   CTDSAM=CTDSAM/FLOAT(NCTLPT)      & IGNORES NON-LINEARITY
   CTDLAT=CTDLAT/FLOAT(NCTLPT)     & IGNORES PARALLEL CURVATURE
   CTDLON=CTDLON/FLOAT(NCTLPT)     & IGNORES MERIDIAN CONVERGENCE
   STMCHD=CTDLON
   CALL PCTCOV(PCTCTL)
C
C SAVE NOMINAL ALTITUDE
C
   SAMIKM=NERSAM/(ALTKM*(TAN(ROLRAD-SCNTH2)
   & -TAN(ROLRAD-SCNTH2)))
   ALTNOM=ALTKM
C
C ESTIMATE CORRECTED LINE AND SAMPLE NUMBERS AND STM COORDINATES
C
   CALL TAPCOR
C
C ESTIMATE FORWARD STM COEFFICIENTS
C
   CALL DLSTSG(INETHI,NETPT,
   & CORNET(WLIN,1),CORNET(WSAM,1),
   & STMNET(WNO,1),STMNET(WEA,1),CORSTM)
```

```

C
C
C COMPUTE FINAL ALTITUDE AND PROJECTION CENTRAL MERIDIAN
C
  IF(NERGEO.EQ.'ERT') CALL SATALT(ALTHM,SAMKM)
  CALL Q4(CTRLAT,CTRLON,CTRLIN,CTRSAM)
  STMCHD=AIN(0.5*100.*CTRLON)/100.
  IF(NERGEO.EQ.'UTM') STMCHD=((FIX(CTRLON)+5)/6)*6-3  & CLOSEST UTM C.M.

C
C COMPUTE FINAL CORRECTED LINE AND SAMPLE NUMBERS AND STM COORDINATES
C
  CALL TAPCOR

C
C COMPUTE FINAL FORWARD & INVERSE STM COEFFICIENTS
C
  CALL DLSTSQ(NETH1,NETPT,
  & CORNET(MLIN,1),CORNET(WSAM,1),
  & STMNET(MNO,1),STMNET(WEA,1),CORSTM)
  CALL REVERT(CORSTM,STMCOR)

C
C COMPUTE STM/GEOGRAPHIC COORDINATES OF SCENE CENTER AND NADIR
C
  CTRN=STMN4C(CTRLIN,CORS4(CTRLIN,CTRSAM))
  CTRE=STME4C(CTRLIN,CORS4(CTRLIN,CTRSAM))
  CALL Q4(CTRLAT,CTRLON,CTRE,CTRN,STMCHD)
  DIRLAT=0.      & FUTURE COMPUTATION
  DIRLON=0.      & FUTURE COMPUTATION

C
C COMPUTE STM/GEOGRAPHIC COORDINATES OF CONTROL NETWORK CENTROID
C
  CTDN=STMN4C(CTDLIN,CORS4(CTDLIN,CTDSAM))
  CTDE=STME4C(CTDLIN,CORS4(CTDLIN,CTDSAM))
  CALL Q4(CTDLAT,CTDLON,CTDE,CTDN,STMCHD)

C
C COMPUTE DIFFERENCE BETWEEN SCENE CENTER & CONTROL NETWORK CENTROID
C
  DCLIN=CTRLIN-CTDLIN
  DCSAM=CTRSAM-CTDSAM
  DCLAT=CTRLAT-CTDLAT
  DCLON=CTRLON-CTDLON
  DCKM=.001*SQRT((CTRE-CTDE)**2+(CTRN-CTDN)**2)

C
C COMPUTE ANGLE OF ROTATION FROM STM TO SRM
C (EASTING AXIS OF SRM IS PARALLEL TO MSS SCAN LINES)
C
  ROTRAD=ATAN(CORSTM(2)/CORSTM(5))

C
C COMPUTE 1:1 MAPPING COEFFICIENTS IN METERS
C
  SRMLN=CORSRM(1)*CORLIN+CR(2)*CORSAM+CR(3)
  
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CONADJ
000

```
C          SRNSAM=CORSRH(4)*CORLIN+CR(5)*CORSAM+CR(6)
C
C          CORSRH(1)=CORSTH(4)*SIN(ROTRAD)-CORSTH(1)*COS(ROTRAD)
C          CORSRH(2)=0.
C          CORSRH(3)=0.
C          CORSRH(4)=CORSTH(4)*COS(ROTRAD)+CORSTH(1)*SIN(ROTRAD)
C          CORSRH(5)=CORSTH(5)*COS(ROTRAD)+CORSTH(2)*SIN(ROTRAD)
C          CORSRH(6)=0.
C
C          PRINT ADJUSTMENT SUMMARY
C
C          WRITE(6,615)
C          1 CTRLIN,CTRSAM,CTRLAT,CTRLON.
C          2 CTOLIN,CTOSAM,CTOLAT,CTOLON.
C          3 DCLIN,DCSAM,DCLAT,DCLON,DCKM
C          615 FORMAT('0',20X,'LINE SAMPLE LATITUDE LONGITUDE'/
C          1 ' SCENE CENTER      '.2(F8.2,1X),2(F10.5,1X)/
C          2 ' CONTROL CENTROID '.2(F8.2,1X),2(F10.5,1X)/
C          3 ' CENTER-CENTROID '.2(F8.2,1X),2(F10.5,1X).
C          4 ' ('.F5.1,' KM)')
C          WRITE(6,625) STNCHD
C          625 FORMAT(' PROJECTION CENTRAL MERIDIAN'.18X,F10.5)
C          WRITE(6,635) PCTCTL
C          635 FORMAT('CONTROL COVERAGE  '.F6.1,' PERCENT')
C          WRITE(6,645) ALTNOM
C          645 FORMAT('NOMINAL ALTITUDE  '.F6.1,' KM')
C          WRITE(6,655) ALTKM
C          655 FORMAT(' COMPUTED ALTITUDE  '.F6.1,' KM')
C
C          COMPUTE AND PRINT RESIDUAL ERRORS
C
C          CALL ERRORS(RHSMET)
C
C          STORE REGISTRATION PARAMETERS
C
C          CALL STREOB
C          NHNDOM=NHNDOW+1
C
C          RESTORE NOMINAL ALTITUDE
C
C          ALTKM=ALTNOM
C
C          NORMAL RETURN
C
C          900 KOND='
C          RETURN
C
C
C
C
C
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CONADJ
887

```
C
C   INTERNAL
C   SUBROUTINE TAPCOR
C
C   CALL TRACE
ALTSAM=ALTKM*SAIKM
DO 100 K = 1,NETM
CORNET(WLIN,K)=CORLVA(ADJNET(WLIN,K),ADJNET(WSAM,K))
CORNET(WSAM,K)=CORSAVA(ADJNET(WLIN,K),ADJNET(WSAM,K))
IF(NERTS(3).LT.0) CORNET(WSAM,K)=ADJNET(WSAM,K) & NO NON-LINEAR CORRECTION
CALL U401
& STMNET(WEA,K),STMNET(WNU,K),
& OEDNET(WLAT,K),OEDNET(WLON,K),STMCHD)
100 CONTINUE
RETURN
```

```
C
C
C
C
C
C
C   INTERNAL
C   SUBROUTINE SATALT(ALTKM,SAIKM)
C
C   CALL TRACE
BEON=STMN4C(.5*NERLIN,1.)
BEOE=STMN4C(.5*NERLIN,1.)
ENDN=STMN4C(.5*NERLIN,NEERSAM)
ENDE=STMN4C(.5*NERLIN,NEERSAM)
BASEKM=.001*SQRT((BEOE-ENDE)**2 + (BEON-ENDN)**2)
ALTKM=BASEKM/(TAN(ROLRAD*SCNTH2)-TAN(ROLRAD*SCNTH2))
SAIKM=FLOAT(NEERSAM-1)/BASEKM
RETURN
```

```
C
C
C
C
C
C   INTERNAL
C   SUBROUTINE ERRORS(RMSMET)
REAL RLCOR(2),RSCOR(2),RLAT(2),RLON(2),RUTH(2)
CALL TRACE
```

```
C
C
C INITIALIZE
C
DO 160 K=1,2
RLCOR(K)=0.
RSCOR(K)=0.
RLAT(K)=0.
RLON(K)=0.
RUTH(K)=0.
160 CONTINUE
WRITE(6,215)
215 FORMAT(
```

1 'RESIDUAL ERRORS IN CONTROL POINT ADJUSTMENT:'/
 2 'OPOINT LINE SAMPLE LATITUDE LONGITUDE'.
 3 ' METERS POINT')

C
 C
 C
 C

COMPUTE RESIDUAL ERRORS

```

DO 806 K=1,NETHI
  KRL=1
  NPRINT=NPRPOS
  NODTYP=' '
  IF(NETPT(K).GT.0) GO TO 802
  KRL=0
  NPRINT=NPRNEG
  NODTYP='CHK'
802 ELCOR=CORNET(WLIN,K)-CORL4S(STMNET(WEA,K),STMNET(WNO,K))
  ESCOR=CORNET(WSAM,K)-CORS4S(STMNET(WEA,K),STMNET(WNO,K))
  RLCOR(KRL+1)=RLCOR(KRL+1)+ELCOR**2
  RSCOR(KRL+1)=RSCOR(KRL+1)+ESCOR**2
  CMPE=STME4C(CORNET(WLIN,K),CORNET(WSAM,K))
  CMPN=STMN4C(CORNET(WLIN,K),CORNET(WSAM,K))
  CALL G4U(CMPLAT,CMPLON,CMPE,CMPN,STMCMD)
  ELAT=GEDNET(WLAT,K)-CMPLAT
  ELGN=GEDNET(WLON,K)-CMPLON
  EUTH=(STMNET(WEA,K)-CMPE)**2 + (STMNET(WNO,K)-CMPN)**2
  RLAT(KRL+1)=RLAT(KRL+1)+ELAT**2
  RLOK(KRL+1)=RLOK(KRL+1)+ELON**2
  RUTH(KRL+1)=RUTH(KRL+1)+EUTH
  EUTH=SQRT(EUTH)
  IF(NPRINT.NE.0) WRITE(6,804)
  1 NETPT(K),ELCOR,ESCOR,ELAT,ELON,
  2 EUTH,NODTYP,NETPT(K)
804 FORMAT(2X,I4,2X,2(F7.2,1X),2(F10.5,1X),
  2 F7.1X,A3,I4)
806 CONTINUE

```

C
 C
 C
 C

COMPUTE ROOT MEAN SQUARE ERRORS

```

NODTYP='CTL'
N=NCTLPT
KRL=1
807 RLCOR(KRL+1)=SQRT(RLCOR(KRL+1)/N)
  RSCOR(KRL+1)=SQRT(RSCOR(KRL+1)/N)
  RLAT(KRL+1)=SQRT(RLAT(KRL+1)/N)
  RLOK(KRL+1)=SQRT(RLOK(KRL+1)/N)
  RUTH(KRL+1)=SQRT(RUTH(KRL+1)/N)
  WRITE(6,809) NODTYP,RLCOR(KRL+1),RSCOR(KRL+1),
  1 RLAT(KRL+1),RLOK(KRL+1),RUTH(KRL+1),N,NODTYP
809 FORMAT('RMS ',A3,2(F7.2,1X),2(F10.5,1X),F9.1,1X,13,1X,A3,' PTS')
  IF(KRL.EQ.0) GO TO 810
  IF(NCTLPT.EQ.NETHI) GO TO 810      a NO CHECK POINTS
  NODTYP='CHK'
  N=NETHI-NCTLPT      a NUMBER OF CHECK POINTS
  KRL=0
  GO TO 807

```

DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

CONADJ
 009

```

010 RMSNET=RUTH(1+1)      & CONTROL POINT RMS (KRL=1)
    WRITE(8,045)
045 FORMAT(1X)
    RETURN
  
```

C
 C
 C
 C
 C
 C
 C

```

INTERNAL
SUBROUTINE PCTCOV(PCTCTL)
  
```

C THIS INTERNAL SUBROUTINE ESTIMATES THE PERCENT OF THE SCENE COVERED BY
 C THE CONTROL NETWORK. THE AREA OF THE NETWORK IS APPROXIMATED BY THE AREA
 C OF AN ELLIPSE WITH RADII EQUAL TO GEOMETRIC MEAN RADII OF THE TWO SMALLEST
 C ROTATED ENVELOPES.

```

REAL RLS(4,8),DLSMIN(2),DLSMAX(2)
REAL P1/3.14159285/
CALL TRACE
  
```

C
 C
 C
 C

C INITIALIZE

```

DO 140 N=1,31.2
  RLS(N,1)=+999999.      & INITIAL MINIM.
140 RLS(N+1,1)=-999999.  & INITIAL MAXIMA
  DLSMIN(1)=+999999.
  DLSMIN(2)=+999999.
  DLSMAX(1)=-999999.
  DLSMAX(2)=-999999.
  
```

C
 C
 C
 C

C FIND ROTATED ENVELOPES

```

DO 280 K=1,NETHI
  IF(NETPT(K).LT.0) GO TO 280
  ROTLIN=ADJNET(WLIN,K)
  ROTSAH=ADJNET(HSAM,K)
  DO 240 NROT=1,8
    RLS(1,NROT)=AMINI( RLS(1,NROT),ROTLIN)
    RLS(2,NROT)=AMAXI( RLS(2,NROT),ROTLIN)
    RLS(3,NROT)=AMINI( RLS(3,NROT),ROTSAM)
    RLS(4,NROT)=AMAXI( RLS(4,NROT),ROTSAM)
    TEMP=ROTLIN*COS(P1/16.)-ROTSAM*SIN(P1/16.)
    ROTSAH=ROTLIN*SIN(P1/16.)+ROTSAM*COS(P1/16.)
    ROTLIN=TEMP
  240 CONTINUE
  280 CONTINUE
  
```

C
 C
 C
 C

C FIND MINIMUM/MAXIMUM ENVELOPE DIMENSIONS

```

DO 320 NR=2,32.2
  DLSMIN(1)=AMINI(DLSMIN(1),(RLS(NR,1)-RLS(NR-1,1)))
  DLSMAX(1)=AMAXI(DLSMAX(1),(RLS(NR,1)-RLS(NR-1,1)))
  320 CONTINUE
  
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CONADJ
010

```
DO 340 NR=2,32.2
  IF((RLS(NR,1)-RLS(NR-1,1)).EQ.DLSMIN(1)) GO TO 330
  DLSMIN(2)=AMIN1(DLSMIN(2),(RLS(NR,1)-RLS(NR-1,1)))
330 IF((RLS(NR,1)-RLS(NR-1,1)).EQ.DLSMAX(1)) GO TO 340
  DLSMAX(2)=AMAX1(DLSMAX(2),(RLS(NR,1)-RLS(NR-1,1)))
340 CONTINUE

C
C
C ESTIMATE RADII/AREA/PERCENT
C
  PIXNW=PI
  & *SQRT(DLSMIN(1)*DLSMIN(2)/4)
  & *SQRT(DLSMAX(1)*DLSMAX(2)/4)
  PIXTOT=FLOAT(NERLIN)*FLOAT(NERSAM)
  PCTCTL=100.*PIXNW/PIXTOT
  RETURN
  END
```



```

      INCLUDE WINDEF.LIST      & DEFINE STRUCTURE OF WINDOW PACKETS (& KONNET)
      INCLUDE NULCST.LIST     & DEFINE NULL CHARACTER STRING
C
C LOCAL DECLARATIONS
C -----
C
      INTEGER NDSAVE          & TEMPORARY SAVE FOR CONTENTS OF NOTOTL ON ENTRY
      INTEGER MIXL(24,36)    & ARRAY OF DIAGRAM SUPER-PIXELS
      INTEGER MATT(29)       & DYNAMIC FORMAT SPECIFICATIONS FOR SUPER-PIXELS
      INTEGER NODTYP(4,3)    & NODE TYPES:
      DATA NODTYP/
C      111111222222333333444444
      1 'CONTROL POINTS ONLY      .'
      2 'CHECK POINTS ONLY      .'
      3 'ALL POINTS              './
C
C
C PROCEDURE
C -----
C
      CALL TRACE
C
C CHECK IF SCENE SIZE AND CENTER ARE DEFINED
C
      NDSAVE=NDTOTL
      IF((NERLIN.LT.100).OR.(NERLIN.GT.10000)) CALL MDWARN(
      - 'SIZE (LINES) NOT DEFINED')
      IF((NERSAM.LT.100).OR.(NERSAM.GT.10000)) CALL MDWARN(
      - 'SIZE (SAMPLES) NOT DEFINED')
      IF((CTRLIN.LT.50.).OR.(CTRLIN.GT.5000.)) CALL MDWARN(
      - 'CENTER (LINE) NOT DEFINED')
      IF((CTRSAM.LT.50.).OR.(CTRSAM.GT.5000.)) CALL MDWARN(
      - 'CENTER (SAMPLE) NOT DEFINED')
      IF(NDSAVE.NE.NDTOTL) GO TO 900
C
C
C DETERMINE WHICH POINTS TO DIAGRAM
C
      NODT='ALL'
      CALL GETSKH(NODT,3,NULCST)
      IF(NODT.EQ.'CON') GO TO 210      & JUST CONTROL POINTS
      IF(NODT.EQ.'CHE') GO TO 220      & JUST CHECK POINTS
      IF(NODT.EQ.'ALL') GO TO 230      & ALL POINTS
      IF(NODT.EQ.'ERR') GO TO 240      & ERROR DIAGRAM
      CALL WARNB('BAD DIAGRAM SPECIFICATION ---')
      GO TO 900
      210 ASSIGN 330 TO IPOS
      ASSIGN 370 TO INEG
      NODT=I
      IF(NCTLPT.GT.0) GO TO 300
      CALL MDWARN('NO CONTROL POINTS')
      GO TO 900
      220 ASSIGN 330 TO IPOS
      ASSIGN 370 TO IPOS
  
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CONDIA
003

```

      NODT=2
      IF((NETHI-NCTLPT).GT.0) GO TO 300
      CALL HDWARN('NO CHECK POINTS')
      GO TO 900
230  ASSIGN 330 TO IPOS
      ASSIGN 330 TO INEQ
      NODT=3
      IF(NETHI.GT.0) GO TO 300
      CALL HDWARN('NO POINTS')
      GO TO 900
240  CALL DIAERR
      GO TO 900
C
C
C INITIALIZE
C
300  NHNDOW=IABS(NHNDOW)
      IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE
      CALL HDUNIT(0.6)
      WRITE(6,305) NERTS.(NODTYP(N,NODT).N=1,4)
305  FORMAT(' SCENE '.11,J4,'-',J5,' -- '.4A6)
      DO 320 I=1,864
320  MIXL(I,1)=0
C
C
C LOAD DIAGRAM ARRAY WITH POINT NUMBERS
C
      DO 370 N=1,NETHI
      IF(NETPT(N))INEQ,370,IPOS
330  INTEMP=IABS(NETPT(N))      & DON'T PRINT SIGN      &-&-&-& FIX &-&-&-&
      INTEMP=MOD(INTEMP,100)      & STRIP OFF HUNDREDS DIGIT &-&-&-& FIX &-&-&-&
      ILMAP=4.+30.*(ADJNET(WLIN,N)-1)/(CTRLIN+NERLIN/2.)
      ILMAP=MAX0(ILMAP,2)
      ILMAP=MIN0(ILMAP,35)
      ISMAP=1.+24.*(ADJNET(WSAM,N)/(CTRSAM+NERSAM/2.))
      ISMAP=MAX0(ISMAP,1)
      ISMAP=MIN0(ISMAP,24)
      IF(MIXL(ISMAP,ILMAP).EQ.0) GO TO 360
      IF(ILMAP.GT.18) GO TO 350
      ILMAP=ILMAP+1
      IF(MIXL(ISMAP,ILMAP).EQ.0) GO TO 360
      ILMAP=ILMAP-2
      IF(MIXL(ISMAP,ILMAP).EQ.0) GO TO 360
      ILMAP=ILMAP+3
      IF(MIXL(ISMAP,ILMAP).EQ.0) GO TO 360
      GO TO 370
350  ILMAP=ILMAP-1
      IF(MIXL(ISMAP,ILMAP).EQ.0) GO TO 360
      ILMAP=ILMAP+2
      IF(MIXL(ISMAP,ILMAP).EQ.0) GO TO 360
      ILMAP=ILMAP-3
      IF(MIXL(ISMAP,ILMAP).NE.0) GO TO 370
360  MIXL(ISMAP,ILMAP)=INTEMP
370  CONTINUE
&-&-&-& FIX &-&-&-&
C
C
```

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CONDIA
004

C DIAGRAM POINTS ABOVE SCENE

C
MATT(01)='(2H .'
MATT(08)='1X.'
MATT(15)='1X.'
MATT(22)='1X.'
MATT(29)='1X)'
ILMAPN=1
MLAX=3
CALL TROLIN

C
C
C DIAGRAM POINTS WITHIN SCENE

C
WRITE(6.505)
505 FORMAT(' ++++++',
1 ' ++++++')
MATT(01)='(2H +.'
MATT(08)='1H+.'
MATT(15)='1H+.'
MATT(22)='1H+.'
MATT(29)='1H+)'
ILMAPN=4
MLAX=33
CALL TROLIN
WRITE(6.505)

C
C
C DIAGRAM POINTS BELOW SCENE

C
MATT(01)='(2H .'
MATT(08)='1X.'
MATT(15)='1X.'
MATT(22)='1X.'
MATT(29)='1X)'
ILMAPN=34
MLAX=36
CALL TROLIN
WRITE(6.505)
585 FORMAT(12X,'(HUNDREDS DIGIT OMITTED)')
NWNDOW=NWNDOW+1

C
C
900 KOMO=''
RETURN

C
C
C
C
SUBROUTINE TROLIN
DO 800 ILMAP=ILMAPN,MLAX
DO 700 ISMAP=1,24
NFI=ISMAP+1+(ISMAP-1)/6
IF(NIXL(ISMAP,ILMAP).EQ.0) GO TO 600
MATT(NFI)='12.'

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CONDIA
005

```
GO TO 700
800 MAT(MF1)='A2.'
    MIXL(ISHAP,ILMAP)='
700 CONTINUE
    WRITE(6,MAT) (MIXL(ISHAP,ILMAP),ISHAP=1,24)
800 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE CONEX1  & TERMINATION ROUTINE FOR CONTROL
      -----
C
C (E H SCHLOSSER)
C
C      INCLUDE KOMXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C      INCLUDE NULCST.LIST     & DEFINE NULL CHARACTER STRING
C
C      CALL TRACE
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
C -----
C
C      PSTOP
C
C
C TERMINATE PROGRAM
C
C      IF(NDFATL.NE.0) CALL PABORT(  NULCST)
C      CALL PSTOP(  NULCST)
C
C
C PSTOP DOES NOT RETURN
C
C      END
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CONXQT
002

```
C
C IDENTIFY PROGRAM
C
C   CALL PSTART( 'DAM CONTROL(0009)')
C
C
C FLAG PITCH & ROLL AS UNKNOWN
C
C   PITDEQ**999999.
C   ROLDEQ**999999.
C
C
C QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE
C
C   300 CALL SYSADD(LOCFIL, 'MACDAM','DEF-CONTROL',,')
C     IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL, 'DAM','DEF-CONTROL',,')
C     IF(LOCFIL.LE.0) CALL MDFATL( 'NO DEFAULT COMMANDS')
C
C
C ON RETURN, CALL CON000 TO GET DEFAULT/USER COMMANDS
C
C   CALL NVIATO( CON000.NULSUB)
C
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
C   RETURN
C   END
```

```
      SUBROUTINE DIAERR  & DIAGRAM CONTROL/CHECK POINT ERRORS
      -----
C
C
C (H L BROWN)
C
C
      DIMENSION JWORD(99),ITAG(350)
      DATA JWORD/99* ' /
      INCLUDE KOMFIT.LIST
      INCLUDE KOMNER.LIST
      INCLUDE WINDEF.LIST
      INCLUDE ASHDEF.LIST
      INCLUDE KOMNET.LIST
      INCLUDE TRFORM.LIST
      INCLUDE KOMXQT.LIST
      DEFINE IFXRND(REAL)=IFIX(REAL*SIGN(.5,REAL))
      CALL TRACE
C
C
C THIS ROUTINE DIAGRAMS ERRORS IN BOTH
C CONTROL AND CHECK POINTS.
C
C
C DETERMINE IF MODE IS BATCH OR DEMAND
C
      INDE=102
      IF(MBATCH.EQ.0) INDE=35
      IF(INDE.EQ.102) GO TO 202
C
C
C SET PLOT CONSTANTS FOR DEMAND MODE
C
      BETA=18.0
      ICTR=18
      NCOL=33
      GO TO 204
C
C
C SET PLOT CONSTANTS FOR BATCH MODE
C
202 BETA=48.
      ICTR=48
      NCOL=98
204 ITEMP='JUNK'
C
C
C DETERMINE IF LINE OR SAMPLE ERROR TO BE PLOTTED
C
      CALL GETSKN(IITEMP,3,-0)
      IF(IITEMP.EQ.'LIN') GO TO 210
      IF(IITEMP.EQ.'SAM') GO TO 220
      CALL WARNB('BAD ERROR AXIS --')
      GO TO 900
210 LSPRT=1
      JERRAX='LINE'
      GO TO 300
```


DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

OIAERR
888

```
220 LSPRT=2
    JERRAX='SAMPLE'
C
C
C SET UP SORT
C
300 CALL GETSKH(ITEMP,3,-0)
    IF(ITEMP.EQ.'SOR') GO TO 350
    CALL WARN9('SORT NOT SPECIFIED ---')
    GO TO 900
350 CALL GETSKH(ITEMP,3,-0)
    IF(ITEMP.EQ.'LIN') GO TO 410
    IF(ITEMP.EQ.'SAM') GO TO 420
    CALL WARN9('BAD SORT AXIS ---')
    GO TO 900
410 LSSRT=1
    JSRTAX='LINE'
    GO TO 450
420 LSSRT=2
    JSRTAX='SAMPLE'
C
C
C CHECK IF NETWORK HAS BEEN ADJUSTED
C
450 IF(IRMSNET.NE.0) GO TO 500
    CALL MDWARN('NETWORK NOT ADJUSTED')
    GO TO 900
500 DO 600 K=1,NETH1
    ITAG(K)=(FIX(ADJNET(LSSRT,K)*2)+20)
    ASHMZ(ITAG(K))=K
600 CONTINUE
C
C
C BEGIN SORT
C
    CALL ISRTNA(ITAG,NETH1)
    IF(INDX.EQ.35) GO TO 610
    GO TO 614
610 WRITE(6,605) JERRAX,JSRTAX
605 FORMAT('1POINT'.3X,'ERROR'.3X,'SORT'.6X,
     1'-2.0'.12X,'0'.11X,'2.0',
     2/9X,A6,2X,A6/)
    GO TO 617
614 WRITE(6,615) JERRAX,JSRTAX
615 FORMAT('1POINT'.3X,'ERROR'.3X,'SORT'.7X,
     1'-2.0'.44X,'0'.44X,'2.0',
     2/9X,A6,2X,A6/)
617 IM=2.0
C
C
C CALCULATE AND PLOT THE ERRORS
C
    DO 750 K=1,NETH1
    KK=ASHMZ(ITAG(K))
    NODTYP='*'
    IF(NETPT(KK).GT.0) GO TO 702
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DIAERR
003

```
      NODTYP='.'
702 IF (LSPRT.EQ.1) GO TO 704
      ERR=CORNET(MSAH, KK) - CORN4S(STHNET(MEA, KK), STHNET(MNO, KK))
      GO TO 705
704 ERR=CORNET(MLIN, KK) - CORL4S(STHNET(MEA, KK), STHNET(MNO, KK))
705      J=IFXRND(BETA*(AMIN*(+2., AMAX)*(-2., ERR)))/(M*2.0)-BETA
      JWORD(CTR)='.'
      JWORD(J)=NODTYP
      WRITE(6,710) NETPT(KK), ERR, ADJNET(LSSRT, KK), (JWORD(N), N=1, NCOL)
710  FORMAT(1X, 14.2X, F5.2, 2X, F7.2, 7X, 102A1)
      JWORD(J)='.'
750 CONTINUE
900 RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE DLSTSQ(NODES,IPT,X,Y,P,Q,A)
C (W O EPPLER / E H SCHLOSSER)
C THIS SUBROUTINE COMPUTES A(1) THRU A(6) TO MINIMIZE (P-U) AND (Q-V)
C WHERE U=A(1)*X+A(2)*Y+A(3)
C AND V=A(4)*X+A(5)*Y+A(6)
C
C THIS SUBROUTINE CALLS DGJR TO SOLVE THE SIMULTANEOUS EQUATIONS REPRESENTED BY
C MATRIX H (DGJR IS IN THE UNIVAC MATHPACK LIBRARY)
C
      DOUBLE PRECISION H,V
      DIMENSION IPT(1),X(1),Y(1),Z(700),P(1),Q(1),H(3,5),A(6),JC(1),V(2)
      DEFINE I(IPT(1))-IPT(1)
      CALL TRACE
C
      NTOT=NODES*2
      NCTLPT=0
      XB=0.
      YB=0.
      PB=0.
      QB=0.
      DO 1 K=1,NTOT,2
      IF(I(IPT(K/2+1)).LT.0) GO TO 1
      XB=XB+X(K)
      YB=YB+Y(K)
      PB=PB+P(K)
      QB=QB+Q(K)
      NCTLPT=NCTLPT+1
1  CONTINUE
      XB=XB/NCTLPT
      YB=YB/NCTLPT
      PB=PB/NCTLPT
      QB=QB/NCTLPT
      DO 2 K=1,NTOT,2
      IF(I(IPT(K/2+1)).LT.0) GO TO 2
      X(K)=X(K)-XB
      Y(K)=Y(K)-YB
      Z(K)=1.
      P(K)=P(K)-PB
      Q(K)=Q(K)-QB
2  CONTINUE
      H(1,1)=DAIP(X,X,IPT,NTOT)
      H(1,2)=DAIP(X,Y,IPT,NTOT)
      H(1,3)=DAIP(X,Z,IPT,NTOT)
      H(2,1)=H(1,2)
      H(2,2)=DAIP(Y,Y,IPT,NTOT)
      H(2,3)=DAIP(Y,Z,IPT,NTOT)
      H(3,1)=H(1,3)
      H(3,2)=H(2,3)
      H(3,3)=DAIP(Z,Z,IPT,NTOT)
      H(1,4)=DAIP(P,X,IPT,NTOT)
      H(2,4)=DAIP(P,Y,IPT,NTOT)
      H(3,4)=DAIP(P,Z,IPT,NTOT)
      H(1,5)=DAIP(Q,X,IPT,NTOT)
      H(2,5)=DAIP(Q,Y,IPT,NTOT)
      H(3,5)=DAIP(Q,Z,IPT,NTOT)
      V(1)=4
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DLSTSQ
002

```
CALL DGJR(H.5.3.3.5.S7.JC.V)
A(1)=M(1.4)
A(2)=M(2.4)
A(3)=M(3.4)-A(1)*XB-A(2)*YB+PB
A(4)=M(1.5)
A(5)=M(2.5)
A(6)=M(3.5)-A(4)*XB-A(5)*YB+QB
DO 3 K=1,NTOT,2
IF(1IPT(K/2+1).LT.0) GO TO 3
X(K)=X(K)+XB
Y(K)=Y(K)+YB
P(K)=P(K)+PB
Q(K)=Q(K)+QB
3 CONTINUE
GO TO 9
7 WRITE(6,8) JC(1)
8 FORMAT(' OVERFLOW AFTER ROW'.I2)
9 RETURN
```

C
C
C
C
C

```
FUNCTION DAIP(R,S,IPT,NTOT)
DIMENSION R(1),S(1),IPT(1)
DOUBLE PRECISION SUM
SUM=0.
DO 1 K=1,NTOT,2
IF(1IPT(K/2+1).LT.0) GO TO 1
SUM=SUM+DBLE(R(K))*DBLE(S(K))
1 CONTINUE
DAIP=SUM
RETURN
END
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLASSIFY
001

C PROGRAM CLASSIFY
C -----
C

C HISTORY
C -----
C

C	E H SCHLOSSER	LEC	07/02/73	ORIGINAL CODE
C	E H SCHLOSSER	LEC	09/24/75	NVIATO/VIATO MEMORY MANAGEMENT
C	E H SCHLOSSER	LEC	11/13/79	UPGRADE DOCUMENTATION

C METHOD
C -----
C

C THIS PROGRAM CLASSIFIES DATA FROM A LANDSAT 'X' OR 'AM' OR 'PM'
C HSS TAPE ASSIGNED TO UNIT 3 AND STORES THE CLASSIFIED BUT UNRESAMPLED
C DATA IN A RANDOM ACCESS DETECTION FILE FOR USE IN THE SAME OR
C SUBSEQUENT RUNS BY OTHER PROGRAMS IN THE DAM PACKAGE.

C THE USER SPECIFIES THE NAME AND SPECTRAL LIMITS FOR ONE MATERIAL,
C A WINDOW DEFINING THE AREA TO BE CLASSIFIED, AND THE TYPE OF DETECTION
C FILE TO GENERATE.

C MACHINE-DEPENDENT CODE
C -----
C

C NONE.

C EXTERNAL REFERENCES
C -----
C

C	NVIATO	& NAME	'VIA' 'TO' ROUTINES
C	VIATO	& CALL	'VIA' 'TO' ROUTINES
C		VIA	TO
C	EXTERNAL	CLA000.	CLAXQT

C EXCEPTIONS
C -----
C

- C 1. IF CONTROL HAS NOT BEEN EXECUTED IN THE CURRENT RUN PRIOR TO
C CLASSIFY AND SATISFACTORILY ADJUSTED A CONTROL NETWORK FOR THE
C SCENE TO BE PROCESSED BY CLASSIFY, THEN CLASSIFY WILL REFUSE TO
C PROCESS THE SCENE AND WILL GENERATE A 'FATAL ERROR.'
- C 2. IF CLASSIFY ENCOUNTERS ANY FATAL ERRORS (SUCH AS FATAL TAPE PARITY/
C POSITIONING ERRORS) IT WILL ERROR TERMINATE AND FLAG THE OUTPUT
C DETECTION FILE AS DEFECTIVE. THE RUN WILL NOT BE ABORTED, BUT
C SUBSEQUENT EXECUTIONS OF PRDENS/PRTCLASS/PLTCLASS WILL BE ALERTED
C TO THIS FATAL ERROR.
- C 3. IF CLAXQT DOES NOT CALL NVIATO TO CHANGE THE 'VIA' AND/OR 'TO'
C ROUTINES, THEN CLASSIFY WILL CALL TO CLAXQT IN AN ENDLESS LOOP!

DAW PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLASSIFY
002

C
C
C GLOBAL DECLARATIONS
C -----
C

INCLUDE KOMXQT.LIST	COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMLOG.LIST	COMMON LOG FILE BUFFER, I/O PKT. POINTERS
INCLUDE KOMLUS.LIST	COMMON POINTERS/FLAGS FOR UNIT 3
INCLUDE KOMLUS.LIST	COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
INCLUDE KOML2N.LIST	COMMON I/O PKTS FOR DETECTION FILES (UNITS 2N
INCLUDE KOMINH.LIST	COMMON INPUT WINDOW PACKETS
INCLUDE KOMOHV.LIST	COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMNER.LIST	COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST	COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST	COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMIRT.LIST	COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
INCLUDE KOMDET.LIST	COMMON DETECTION FILE WINDOW PACKETS & DATES
INCLUDE KOMSLM.LIST	COMMON SPECTRAL LIMITS
INCLUDE KOMALT.LIST	COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS

C
C
C PROCEDURE
C -----
C

```
CALL NVIATO( CLA000.CLAXQT)  ; FIRST CALL IS VIA CLA000 TO CLAXQT
100 CONTINUE
    CALL VIATO
GO TO 100
END      ; (STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE)
```


PROGRAM CLASSIFY/VIRTUAL

HISTORY

E H SCHLOSSER LEC 08/02/74 ORIGINAL CODE
E H SCHLOSSER LEC 11/08/79 3MAP.FZ(IN): NO 'N' IN DEMAND)

METHOD

CONSTRUCT 3MAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPF5.
CONSTRUCT 3XQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPF5.
WRITE 3MAP & 3XQT COMMANDS TO TEMPORARY FILE 20.
3ADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 6-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS 3 FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IOFS 3 INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS 3 TERMINATE PROGRAM EXECUTION
DAM.CLASSIFY-MAP 3 SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAOPT 3 STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS 3ASO-D & 3PREP-D.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

(2 = REAL TIME
(3 = LOW EXEC
(4 = DEMAND
(5 = DEADLINE BATCH
(6 = BATCH

(3XQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER A5 IN

(MASTER BIT NOTATION.

LOCAL DECLARATIONS

```

      AXRS
S(00) . D-BANK
SSSH      FORM      6.6.6.18
          111111222222333333444444555555666666777777888888999999
LABSDF    SSSH      050.1.'F'.0      . LABEL. 1 WD. FORTRAN. FIELDATA
LABIMG    *SDF*
MAPSDF    SSSH      000.9.0.0      . DATA. 9 WDS. . FIELDATA
MAPIMG    *XQTS: MAP.FZN DAM.CLASSIFY-MAP.CLASSIFY . :XQTS*
AODSDF    SSSH      000.9.0.0
ADDIMG    *XQTS: ADD      DAM.SYS-MAPOPT . :XQTS*
XQTSDF    SSSH      000.9.0.0
XQTIMG    *XQTS: XQT.1      CLASSIFY . :XQTS*
EOFSDF    -      0      . END-OF-FILE STOP WORD
PF        FORM      12.6.18
CSFASG    *ASG.T 20. . .
CSFADD    *ADD 20. . .
SAVREG    RES      1
IOPKT     ISOO     '20'.WS 33.LABSDF.'0' 0
  
```

PROCEDURE

```

S(01) . I-BANK
CLASSIFY* LA.U      A0.'      . A0 := .
          TNE.U     A4.4      . SKIP NEXT INST IF A4<>4 (NOT DEMAND)
          SA.S2     A0.MAPIMG+2 . DEMAND! BLANK OUT N OPTION
          LA        A0.(CSFASG) . ADDRESS OF ASG IMAGE
          ER        CSFS      . DO IT
          SA        A0.SAVREG  . STORE &
          P$PRINT   (PF 2.1.SAVREG) . PRINT ASG STATUS

          GETOPT     . LOAD OPT LTRS INTO A2.A3.A4

          PUTOPT    DS        A2.XQTIMG+2 . STORE OPTION LETTERS INTO XQTI IMAGE
          SA        A4.XQTIMG+4 . (3 WORDS -- MAX 18 OPT LETTERS)

          WRITE     LA        A0.(IOPKT) . ADDRESS OF I/O PACKET
          ER        IOWS      . WRITE SDF IMAGES TO 20.

          ADD       LA        A0.(CSFADD) . ADDRESS OF ADD IMAGE
          ER        CSFS      . DO IT
          ER        EXITS

          END        CLASSIFY
  
```

CLASSIFY OVERLAY STRUCTURE

HISTORY

E H SCHLOSSER	LEC	08/23/74	ORIGINAL CODE
E H SCHLOSSER	LEC	07/14/78	CHANGE OVERLAYS TO REDUCE THRASHING
E H SCHLOSSER	LEC	10/30/78	TOLERANCE COMMAND
E H SCHLOSSER	LEC	01/30/79	MACRO COMMANDS & TIME COMMAND
E H SCHLOSSER	LEC	12/18/79	PEEK, POKE, IF, FI
MARY TOMPKINS	LEMSCO	01/11/80	CHANGE OVERLAYS

LIB DAM.

SEG S-MAIN

IN DAM.CLASSIFY/ . MAIN PROGRAM
IN DAM.NVIATO . NAME/CALL 'VIA' AND 'TO' SUBROUTINES
IN DAM.NULSUB . DO NOTHING
IN DAM.SYS-BLOCK . BLOCK DATA SUBROUTINE

UTILITIES FOR MAIN -----

SEG S-FLINFO*(S-MAIN)

IN DAM.FLINFO . GET FILE DESCRIPTIVE INFORMATION

SEG S-R3TASC*(S-MAIN)

IN DAM.CST4AS . CHARACTER STRING FOR ASCII
IN DAM.R3TREC . READ ONE RECORD FROM TAPE (UNIT 3)

MONITOR FOR PHASE 0.1.2.9 COMMANDS -----

SEG S-CLA0129*(S-FLINFO,S-R3TASC)

IN DAM.CLA000 . CALL USER-SPECIFIED PHASE 0 ROUTINE
IN DAM.CLA129 . CALL PREVIOUSLY NAMED PHASE 1/2/9 'TO' ROUTINE
IN DAM.NTABS/DAM . DAM UNIT # TABLE GOES IN SAME SEG W/ FORTRAN I/O

UTILITIES FOR PHASE 0.1.2.9 COMMANDS -----

SEG S-READ5*(S-CLA0129)

IN DAM.READ5 . 'READ' INTO UNIT 5 BUFFER
IN DAM.GET5 . 'GET' FREE-FORMAT FIELD FROM UNIT 5 BUFFER
IN DAM.WARN5 . PROCESS WARNING DIAGNOSTIC FOR UNIT 5 FIELD
IN DAM.SPANS . ENABLE/DISABLE SPANNING FOR UNIT 5

SEG S-OPNCLPR*(S-CLA0129)

IN DAM.OPRCLA . OPEN ALT PRT FILE
IN DAM.CLOSPR . CLOSE ALT PRT FILE

SEG S-CALCROP*(S-CLA0129)

IN DAM.CALSPA . CALIBRATE SPACING
IN DAM.CALWIN . CALIBRATE WINDOW

IN DAM.CALCHA . CALIBRATE CHANNEL POINTERS

. PHASE 0.1.2.9 COMMANDS (FORTRAN I/O ALLOWED) -----

SEQ S-XQTEX1*(S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.CLAXQT . CLASSIFY INITIALIZATION ROUTINE
IN DAM.CLAEX1 . CLASSIFY TERMINATION ROUTINE

SEQ S-PSTART*(S-XQTEX1)
IN DAM.PSTART . GENERAL INITIALIZATION ROUTINE

SEQ S-OPEN3*(S-XQTEX1)
IN DAM.OPENS . OPEN INPUT SCAN DATA FILE (UNIT 3)

SEQ S-OP3DSK*(S-OPEN3)
IN DAM.OP3DSK . OPEN INPUT -- DISK IN PXBDEF FMT (UNIT 3)

SEQ S-OP3BIP*(S-OPEN3)
IN DAM.OP3BIP . OPEN MSS DATA IN BIP FMT (UNIT 3)

SEQ S-OP3MOP*(S-OPEN3)
IN DAM.OP3MOP . OPEN MSS DATA IN MOP FMT (UNIT 3)

SEQ S-03TORHDR*(S-OP3MOP)
IN DAM.03TOR . MOP FMT TAPE DIRECTORY RECORD (UNIT 3)
IN DAM.03HDR . MOP FMT HEADER RECORD (UNIT 3)

SEQ S-03ANOT*(S-OP3MOP)
IN DAM.03ANOT . MOP FMT ANNOTATION RECORDS (UNIT 3)

SEQ S-03SZAM*(S-03ANOT)
IN DAM.03SZAM . SIZE AND INPUT WINDOW FOR AM TAPES

SEQ S-03SZPH*(S-03ANOT)
IN DAM.03SZPH . SIZE AND INPUT WINDOW FOR PH TAPES

SEQ S-03SZAR*(S-03ANOT)
IN DAM.03SZAR . SIZE AND INPUT WINDOW FOR AR TAPES

SEQ S-03SZPR*(S-03ANOT)
IN DAM.03SZPR . SIZE AND INPUT WINDOW FOR PR TAPES

SEQ S-03ANCL*(S-OP3MOP)
IN DAM.03ANCL . MOP FMT ANCILLARY RECORDS (UNIT 3)

SEQ S-LDREG8*(S-XQTEX1)
IN DAM.LDREG8 . LOAD REGISTRATION PARAMETERS FROM UNIT 8

SEQ S-CLOSTOP*(S-XQTEX1)
IN DAM.CLOSE3 . CLOSE INPUT SCAN DATA FILE (UNIT 3)
IN DAM.CLSO2N . CLOSE OUTPUT DETECTION FILE (UNIT 2N)
IN DAM.PSTOP . GENERAL TERMINATION ROUTINE

SEQ S-HELP*(S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KMOCLE . CLEAR WARNINGS/ERRORS

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLASSIFY-MAP
003

IN DAM.KMDEXP . EXPLAIN PROGRAM/COMMAND
IN DAM.KMDFI . END IF...FI BLOCK
IN DAM.KMDIF . BEGIN IF...FI BLOCK
IN DAM.KMONEH . PRINT NEWS
IN DAM.KMONEH . CONDITIONALLY PERFORM NEXT COMMAND
IN DAM.KMDOFF . TURN OFF MODE SWITCH(ES)
IN DAM.KMDOON . TURN ON MODE SWITCH(ES)
IN DAM.KMDPEE . PEEK
IN DAM.KMDPOK . POKE
IN DAM.KMDREN . RENUMBER (GET/CHECK NEW WINDOW SEQUENCE NUMBER)
IN DAM.KMDTIM . PRINT CLOCK TIME & CHARGE TIME

SEG S-GEOMETRY*.(S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KMDWIN . GET/CHECK WINDOW ENVELOPE/VERTICES
IN DAM.KMDZON . GET/CHECK UTM PROJECTION ZONE

SEG S-SPECS-9*.(S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KMDCHA . GET/CHECK RAW/TRANSFORMED SCANNER CHANNEL(S)
IN DAM.KMDHEA . GET/CHECK PAGE HEADING(S)
IN DAM.KMDNAM . GET/CHECK TRANSFORMATION/MATERIAL NAME
IN DAM.KMDORI . GET/CHECK WINDOW ORIGIN
IN DAM.KMDRAD . GET/CHECK RADIANCE LIMITS
IN DAM.KMDTOL . GET/CHECK TOLERANCE

SEG S-MISC*.(S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KMOCOP . GET/CHECK NUMBER OF OUTPUT COPIES
IN DAM.KMDLIN . GET/CHECK LINEAR TRANSFORMATION WEIGHTS/GAIN/BIAS
IN DAM.KMDPAO . SKIP TO TOP OF NEXT PAGE
IN DAM.KMDPRI . GET/CHECK PRINTER SPECIFICATIONS
IN DAM.KMDPOL . GET/CHECK POLAR TRANSFORMATION GAIN/BIAS
IN DAM.KMDSHA . GET/CHECK SHARPENING FILTER COEFFICIENTS

SEG S-EXEC*.(S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.KM0XXX . MACRO COMMANDS
IN DAM.KM00AD . DYNAMIC 0ADD
IN DAM.KM00AS . DYNAMIC 0ASO
IN DAM.KM00BR . DYNAMIC 0BRKPT
IN DAM.KM00FR . DYNAMIC 0FREE
IN DAM.KM00LO . DYNAMIC 0LOG

SEG S-CLADET*.(S-READS.S-OPNCLPR.S-CALCROP)
IN DAM.CLADET . DETECT RADIANCE/DENSITY/CLASS (PHASE 0)

. MONITOR FOR PHASE 3.4.5 COMMANDS -----

SEG S-CLA345*.S-CLA0129
IN DAM.CLA345

. UTILITIES FOR PHASE 3.4.5 COMMANDS -----

SEG S-RD3BIP*.(S-CLA345)
IN DAM.RD3BIL . READ MSS DATA IN BIL FORMAT (UNIT 3)
IN DAM.RD3BSQ . READ MSS DATA IN BSQ FORMAT (UNIT 3)

SEG S-RD3BIP*.(S-CLA345)

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLASSIFY-MAP
004

IN DAM.RD3BIP . MSS DATA IN BIP FORMAT (UNIT 3)

SEQ S-RD3SKNUL*.(S-CLA345)

IN DAM.RD3DSK . DATA ON DISK IN PKDEF FORMAT

IN DAM.RD3NUL . SYNTHETIC DATA -- NO UNIT 3

. PHASE 3.4.5 COMMANDS (NO FORTRAN I/O) -----

SEQ S-CLADE3*(S-RD3MOP,S-RD3BIP,S-RD3SKNUL)

IN DAM.CLADE3 . DETECT RADIANCE IN SELECTED CHANNEL

SEQ S-CLADE4*(S-RD3MOP,S-RD3BIP,S-RD3SKNUL)

IN DAM.CLADE4 . DETECT BINARY CLASSIFICATION DENSITY

SEQ S-CLADE5*(S-RD3MOP,S-RD3BIP,S-RD3SKNUL)

IN DAM.CLADE5 . DETECT CLASS USING PARTITION TABLE

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLASSIFY-MAP/VIRTUAL
001

IN DAM.CLASSIFY/VIRTUAL

SUBROUTINE CLA000(& CALL USER-SPECIFIED PHASE 0 SUBROUTINES FOR CLASSIFY
I NAMSUB) & NAME OF SUBROUTINE TO CALL (OR NULSUB)

C
C
C HISTORY
C -----

C	E M SCHLOSSER	LEC	03/27/73	ORIGINAL CODE
C	E M SCHLOSSER	LEC	06/27/78	DELETE RETN K'S & NUMERIC OPTIONS
C	E M SCHLOSSER	LEC	10/30/78	TOLERANCE COMMAND
C	E M SCHLOSSER	LEC	01/30/79	MACRO COMMANDS & TIME COMMA
C	E M SCHLOSSER	LEC	11/30/79	PEEK, POKE, IF, FI

C
C METHOD
C -----

C RETRIEVE NEXT COMMAND, VALIDATE IT, AND CALL ITS SUBROUTINE.

C
C MACHINE-DEPENDENT CODE
C -----

C NONE.

C
C EXTERNAL REFERENCES
C -----

C	READS	& READ PUNCHED CARD OR TERMINAL INPUT
C	GETSAL	& GET ALPHABETIC COMMAND
C	INTEGER ICE	& INTEGER-CHAR-EQUIV FOR CHARACTER
C	WARNS	& PRINT/LOG WARNING MESSAGE
C	CLA...	& DEDICATED SUBROUTINE FOR COMMAND ... (SEE BELOW)
C	KMD...	& COMMON SUBROUTINE FOR COMMAND ... (SEE BELOW)

C
C EXCEPTIONS
C -----

- C 1. A BLANK COMMAND IS IGNORED.
- C 2. AN INVALID COMMAND GENERATES A DIAGNOSTIC.
- C 3. AN END-OF-FILE ON UNIT 5 IS TREATED THE SAME AS THE EXIT COMMAND.

C
C GLOBAL DECLARATIONS
C -----

C INCLUDE NULCST.LIST & DEFINE NULL CHARACTER STRING

C
C LOCAL DECLARATIONS
C -----

```

C
  INTEGER KOMD           & FIRST 3 CHARS OF USER COMMAND (BLANK AFTER DONE)
  INTEGER LSSTAT        & READS STATUS ('EOF' MEANS END-OF-FILE)
  INTEGER KASE          & MODIFIED I-C-E OF FIRST CHAR OF COMMAND

C
C C PROCEDURE
C -----
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
  CALL TRACE
  CALL NANSUB           & CALL TO NULSUB DOES NOTHING

C
C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C
  KOMD=' NUL'          & IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
  CALL READS(LSSTAT, ' ') & FILL BUFFER, BLANK CUE MESSAGE
  IF(LSSTAT.NE.' ') KOMD='EXI'
  IF(KOMD.NE.'EXI') CALL GETSAL(KOMD.(3), NULCST) & GET 3 ALPHA CHARS

C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C
  KASE=ICE(KOMD)-ICE('A')+1 & A TO Z = 1 TO 26

C
C CASE STATEMENT ON MODIFIED ICE OF COMMAND'S FIRST CHARACTER
C
  IF((KASE.LT.1).OR.(KASE.GT.26)) KASE=27 & NOT ALPHA
  GO TO 1
  0 401.402.403.404.405.406.407.408.409.410.
  1 411.412.413.414.415.416.417.418.419.420.
  2 421.422.423.424.425.426.427)
  & .KASE

C
C DETERMINE COMMAND. PERFORM COMMAND. CHANGE KOMD TO BLANK
C
401 CONTINUE &... A
402 CONTINUE &... B
  GO TO 300

C
403 CONTINUE &... C
  IF(KOMD.EQ.'CHA') CALL KMCHA(KOMD) & CHANNEL
  IF(KOMD.EQ.'CLE') CALL KMCLC(KOMD) & CLEAR
  IF(KOMD.EQ.'COP') CALL KMDCOP(KOMD) & COPIES
  GO TO 300

C
404 CONTINUE &... D
  IF(KOMD.EQ.'DET') CALL CLADET(KOMD) & DETECT
  GO TO 303

C
405 CONTINUE &... E

```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLA000
003

```
IF(KOMD.EQ.'EXI') CALL CLAEXI(KOMD)      3 EXIT
IF(KOMD.EQ.'EXP') CALL KMDEXP(KOMD)     3 EXPLAIN
GO TO 800

C
406 CONTINUE 3*** F
IF(KOMD.EQ.'FI ') CALL KMDFI (KOMD)     3 FI (ENDIF)
GO TO 800

C
407 CONTINUE 3*** G
GO TO 800

C
408 CONTINUE 3*** H
IF(KOMD.EQ.'HEA') CALL KMHEA(KOMD)     3 HEADING
GO TO 800

C
409 CONTINUE 3*** I
IF(KOMD.EQ.'IF ') CALL KMDIF (KOMD)     3 IF
GO TO 800

C
410 CONTINUE 3*** J
411 CONTINUE 3*** K
GO TO 800

C
412 CONTINUE 3*** L
IF(KOMD.EQ.'LIN') CALL KMDLIN(KOMD)     3 LINEAR
GO TO 800

C
413 CONTINUE 3*** M
GO TO 800

C
414 CONTINUE 3*** N
IF(KOMD.EQ.'NAM') CALL KMDNAM(KOMD)     3 NAME
IF(KOMD.EQ.'NEW') CALL KMDNEW(KOMD)     3 NEWS
IF(KOMD.EQ.'NEX') CALL KMDNEX(KOMD)     3 NEXT
GO TO 800

C
415 CONTINUE 3*** O
IF(KOMD.EQ.'OFF') CALL KMDOFF(KOMD)     3 OFF
IF(KOMD.EQ.'ON ') CALL KMDON(KOMD)      3 ON
IF(KOMD.EQ.'ORI') CALL KMDORI(KOMD)     3 ORIGIN
GO TO 800

C
416 CONTINUE 3*** P
IF(KOMD.EQ.'PAG') CALL KMOPAG(KOMD)     3 PAGE
IF(KOMD.EQ.'PEE') CALL KMOPEE(KOMD)     3 PEEK
IF(KOMD.EQ.'POK') CALL KMOPOK(KOMD)     3 POKE
IF(KOMD.EQ.'POL') CALL KMOPOL(KOMD)     3 POLAR
IF(KOMD.EQ.'PRI') CALL KMOPRI(KOMD)     3 PRINTER
GO TO 800

C
417 CONTINUE 3*** Q
GO TO 800

C
418 CONTINUE 3*** R
IF(KOMD.EQ.'RAD') CALL CLARAD(KOMD)     3 RADIANCE
GO TO 800
```

C-2

DAW PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLA000
004

```
C
419 CONTINUE 3*** S
   IF(KOMD.EQ.'SHA') CALL KMDSHA(KOMD)      3 SHARPENING
   GO TO 800
C
420 CONTINUE 3*** T
   IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD)      3 TIME
   IF(KOMD.EQ.'TOL') CALL KMDTOL(KOMD)     3 TOLERANCE
   GO TO 800
421 CONTINUE 3*** U
422 CONTINUE 3*** V
   GO TO 800
C
423 CONTINUE 3*** W
   IF(KOMD.EQ.'WIN') CALL KMDWIN(KOMD)     3 WINDOW
   GO TO 800
C
424 CONTINUE 3*** X
425 CONTINUE 3*** Y
   GO TO 800
C
426 CONTINUE 3*** Z
   IF(KOMD.EQ.'ZON') CALL KMDZON(KOMD)     3 ZONE
   GO TO 800
C
427 CONTINUE 3*** NOT ALPHABETIC
   IF(KOMD.EQ.'SAD') CALL KMD0AD(KOMD)     3 SADD
   IF(KOMD.EQ.'SAD') CALL KMD0AD(KOMD)     3 SADD
   IF(KOMD.EQ.'SAS') CALL KMD0AS(KOMD)     3 SASO
   IF(KOMD.EQ.'SBR') CALL KMD0BR(KOMD)     3 SBRKPT
   IF(KOMD.EQ.'SFR') CALL KMD0FR(KOMD)     3 SFREE
   IF(KOMD.EQ.'SLO') CALL KMD0LO(KOMD)     3 SLOG
C
C
C IF COMMAND WAS NOT FOUND. TRY MACRO-COMMAND
C
800 IF(KOMD.NE.' ') KJMD='CLA-'      3 1ST 3 CHARS OF PROG NAME PLUS '-'
   IF(KOMD.NE.' ') CALL KMDXXX(KOMD)  3 MACRO COMMAND HANDLER
C
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
   IF(KOMD.NE.' ') CALL WARNS('INVALID COMMAND --')
C
C
C FORCE ALL FORTRAN I/O ROUTINES INTO SEQ WITH CLA000 (NEVER PERFORMED)
C
   IF(KOMD.EQ.'JUNK') READ(895,895) KMD
895 FORMAT(IX)
C
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
RETURN
END
```

SUBROUTINE CLA129(8 CALL PREV NAMED PH 1/2/9 'TO' ROUTINE FOR CLASSIFY
1 NANSUB) 8 NAME OF SUBROUTINE TO CALL

C
C
C
C
C
C
C

(E H SCHLOSSER)

CALL PREVIOUSLY NAMED SUBROUTINE

CALL TRACE
CALL NANSUB

C
C
C
C

RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY

RETURN
END

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLAS45
001

SUBROUTINE CLAS45: 3 CALL PHASE 3/4/5 SUBROUTINES FOR CLASSIFY
1 NANSUB) 3 NAME OF SUBROUTINE TO CALL

C

C

C (E M SCHLOSSER)

C

C

C CALL PREVIOUSLY NAMED SUBROUTINE

C

CALL TRACE
CALL NANSUB

C

C

C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY

C

RETURN
END

3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS

CONDITION -----	ACTION -----	DIAGNOSTIC -----
C PROCESSING DEFAULT COMMANDS		
(NWINDOW=0)	NONE	WARNING
KLSTYP SPECIFICATION MISSING	KLSTYP='DEN'	NONE
KLSTYP SPECIFICATION INVALID	NONE	WARNING
NOUTCH SPECIFICATION MISSING	NOUTCH=1	NONE
NOUTCH SPECIFICATION < 1	NONE	WARNING
NOUTCH SPECIFICATION > NFINCH	NONE	WARNING
EXTRA SPECIFICATION	NONE	WARNING
DATA/CHECKOUT MODE	'VIA' ROUTINE IS CLAO00	NONE
WARNING(S) OR FATAL ERROR(S)	'VIA' ROUTINE IS CLAO00	NONE
C GLOBAL DECLARATIONS		

INCLUDE KOMXQT.LIST	% COMMON PROGRAM EXECUTION SWITCHES. COUNTERS	
INCLUDE KOMNER.LIST	% COMMON ERTS SCENE PARAMETERS	
INCLUDE KOMKLS.LIST	% COMMON CLASSIFICATION INFO	
INCLUDE WINDEF.LIST	% DEFINE STRUCTURE OF WINDOW PACKETS	
INCLUDE KOMIHW.LIST	% COMMON INPUT WINDOW PACKETS	
INCLUDE KOMOHV.LIST	% COMMON OUTPUT WINDOW PACKETS	
INCLUDE NULCST.LIST	% DEFINE NULL CHARACTER STRING	
C LOCAL DECLARATIONS		

INTEGER INTEMP	% TEMPORARY	
C PROCEDURE		

CALL TRACE		
C GET/CHECK TYPE OF FIRST OUTPUT DETECTION CHANNEL		
IF(NWINDOW.EQ.0) CALL WARN9('INVALID DEFAULT COMMAND')		
KLSTYP='DEN'	% DEFAULT IS DETECTION OF BINARY CLASS DENSITY	
CALL GETSKH(KLSTYP,(3), NULCST)		
IF(KLSTYP.NE.'RAD') GO TO 240		
CALL NVIATO(CLAS45,CLADE3)	% RADIANCE -- NEXT CALL IS TO CLADE3	
GO TO 300		
240 IF(KLSTYP.NE.'DEN') GO TO 250		
CALL NVIATO(CLAS45,CLADE4)	% DENSITY -- NEXT CALL IS TO CLADE4	
GO TO 300		
250 IF(KLSTYP.NE.'CLA') GO TO 280		
CALL NVIATO(CLAS45,CLADE5)	% CLASS -- NEXT CALL IS TO CLADE5	
GO TO 300		
280 CALL WARN9('BAD DETECT TYPE --')		

```

C
C
C GET/CHK NUMBER OF OUTPUT DETECTION CHANNELS & DRAIN SPECS FOR CURRENT COMMAND
C
C 300 NOUTCH=1
C   CALL GETSIN(NOUTCH.
C     * 1,NLIMCH.
C     & 'MORE OUTPUT CHANNELS THAN SPECIFIED BY CHANNEL COMMAND --')
C     IF((KLSTYP.NE.'RAD').AND.
C       & (NOUTCH.NE.1)) CALL MDWARN:
C     * 'MULTI-CHANNEL DEN/CLA DETECTION FILES NOT IMPLEMENTED'
C     CALL GETSIN(TEMP. +1.-1.'EXTRA DETECT SPECIFICATION --')
C
C
C CHECK RADIANCE LIMITS
C
C   IF(LCVLO1.GT.LCVHI1) CALL MDWARN: 'NO RADIANCE LIMITS'
C   IF(MDATA.C.NE.0) GO TO 900      & DATA/CHECKOUT MODE
C
C
C OPEN PRINT FILE IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
C   IF(NWINDOW.LT.0) CALL OPRCLA      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
C   NWINDOW=ABS(NWINDOW)
C   NPAGE=0
C
C
C PLOT SPECTRAL LIMITS & CALIBRATE WINDOW
C
C   CALL SLMCLA
C   CALL CALSPA
C   CALL CALHIN( 2.)
C   CALL CALCHA
C
C
C ELIMINATE ANY PART OF MSA OUTPUT WINDOW OUTSIDE OF MSA INPUT WINDOW
C
C   MSAOWH(WLIN.WMIN)=MAX0(MSAOWH(WLIN.WMIN).MSA1WW(WLIN.WMIN))
C   MSAOWH(WLIN.WMAX)=MIN0(MSAOWH(WLIN.WMAX).MSA1WW(WLIN.WMAX))
C   MSAOWH(WSAM.WMIN)=MAX0(MSAOWH(WSAM.WMIN).MSA1WW(WSAM.WMIN))
C   MSAOWH(WSAM.WMAX)=MIN0(MSAOWH(WSAM.WMAX).MSA1WW(WSAM.WMAX))
C
C
C ANY MSA DATA AVAILABLE FOR OUTPUT WINDOW??
C
C   IF((MSAOWH(WLIN.WMIN).GE.MSAOWH(WLIN.WMAX)).OR.
C     & (MSAOWH(WSAM.WMIN).GE.MSAOWH(WSAM.WMAX))) KLSTYP='NUL'
C   IF(KLSTYP.EQ.'NUL') CALL MDWARN:
C     * 'NO SCAN DATA WITHIN WINDOW'
C
C
C OPEN OUTPUT DETECTION FILE
C
C   IF(INDOTL.NE.0) GO TO 900
C   CALL OPNOZN      & OPEN NEW OUTPUT DETECTION FILE OR CHECK EXISTING ONE
C   IF(KLSTYP.EQ.'OLD') & IF DESIRED DETECTION DATA ALREADY IN 'OLD' FILE
  
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLADET
004

```
      8          CALL NVIATO( CLA000.NULSUB) 8 ... THEN ACCEPT IT!!  
        IF(NALTM.GT.0) CALL MDUNIT( 4,10)  
        IF(NALTM.GT.0) CALL CLSHDO( 10)  
C  
C  
C ANY DIAGNOSTICS??  
C  
900 IF(INDTOTL.EQ.0) GO TO 990      8 FORCE AHEAD!  
    CALL NVIATO( CLA000.NULSUB)  
    IF(INDFATL.EQ.0) GO TO 920  
    CALL MDNOTE!  
    - 'FATAL ERRORS -- NO DETECTION FILE GENERATED'  
    GO TO 990  
920 IF(INDWARN.EQ.0) GO TO 990  
    CALL MDNOTE!  
    - 'PREVIOUS WARNINGS -- NO DETECTION FILE GENERATED'  
    IF(INDATCH.EQ.0) WRITE(6,925)  
925   FORMAT(4X,'**TRY AGAIN!')  
C  
C  
C RETURN FOR CALL TO NAMED SUBROUTINE  
C  
990 KOND=0  
    CALL MDCLR( NULCST)  
    RETURN  
    END
```


SUBROUTINE CLADE3 3 GENERATE RADIANCE DETECTION FILE

C
C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 03/13/79 ORIGINAL CODE WITH 'CHR' BINS
C E H SCHLOSSER LEC 12/26/79 PXBDEF FORMAT BUFFER W/ 'BYT' BINS
C E H SCHLOSSER LEC 05/16/80 MULTI-CHANNEL RAD DETECTION FILES
C
C
C METHOD
C -----
C
C READ MSA PIXEL RADIANCE DATA. SCREEN EACH PIXEL AGAINST SPECTRAL LIMITS
C FOR CURRENT CHANNEL(S). AND POST RADIANCE FOR PIXELS WHICH PASS TO
C DETECTION FILE.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C ASSUMES 4 BYTES PER INTEGER.
C ASSUMES 20 UNIVAC WORDS PER UNIVAC FASTRAND-FORMATTED DISK SECTOR.
C
C EXTERNAL REFERENCES
C -----
C
C
C MOVIST 3 MOVE INTEGER STRING
C GETRAD 3 GET RADIANCE IN ALL SELECTED RAW/TRANSFORMED CHANNELS
C HDNOTE 3 PRINT/LOG/COUNT 'NOTE' DIAGNOSTIC MESSAGE
C HOFATL 3 PRINT/LOG/COUNT 'FATAL ERROR' DIAGNOSTIC MESSAGE
C PXBDMP 3 DUMP PREAMBLE OF PIXEL BUFFER
C GETBYT 3 GET NON-NEGATIVE NUMBER FROM BYTE STRING
C PUTBYT 3 PUT NON-NEGATIVE NUMBER INTO BYTE STRING
C ERIO 3 INITIATE I/O
C ERWAIT 3 WAIT FOR COMPLETION OF PREVIOUSLY INITIATED I/O
C NVIATO 3 NAME 'VIA' 'TO' SUBROUTINES
C INTEGER NB4NI 3 NUMBER OF BYTES FOR NUMBER OF INTEGERS
C DOUBLE PRECISION CBS4CS 3 VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH
C DOUBLE PRECISION CBS4IN 3 VARIABLE-LENGTH CHAR STRING FOR INTEGER
C VIA TO
C EXTERNAL CLAO00. NULSUB
C
C
C EXCEPTIONS
C -----
C
C 1. IF THE NUMBER OF OUTPUT DETECTION CHANNELS REQUESTED BY NOUTCH IS
C GREATER THAN THE NUMBER OF DETECTION BUFFERS AVAILABLE. THEN A
C 'WARNING' IS GENERATED AND THE ROUTINE RETURNS.
C
C 2. 'BADF' OR 'OFL' ON READING INPUT DATA GENERATES A 'FATAL ERROR' AND

THE ROUTINE RETURNS.

C
 C
 C
 C
 C
 C

GLOBAL DECLARATIONS

INCLUDE KONXQT.LIST	COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONNER.LIST	COMMON ERTS SCENE PARAMETERS
INCLUDE KONKLS.LIST	COMMON CLASSIFICATION INFO
INCLUDE KONFIT.LIST	COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE MINDEF.LIST	DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KONINW.LIST	COMMON INPUT WINDOW PACKETS
INCLUDE KOMOHV.LIST	COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMDET.LIST	COMMON DETECTION FILE WINDOWS
INCLUDE KOML2N.LIST	COMMON I/O PACKETS FOR DETECTION FILES (21-24)
INCLUDE KOMIO.LIST	COMMON I/O FUNCTIONS
INCLUDE KOMSLM.LIST	COMMON CLASSIFICATION SPECTRAL LIMITS
INCLUDE PXBDEF.LIST	DEFINE PIXEL BUFFER STRUCTURE
INCLUDE MAXINT.LIST	DEFINE MAXIMUM INTEGER

C
 C
 C
 C
 C

LOCAL DECLARATIONS

PARAMETER NMBUFS = 8	NUMBER OF MSA PIXEL BUFFERS IN ARRAY
INTEGERS IN MSA BUF = INTS PREAMBLE + (NBINS+3)/4 + (EXTRA BYTES+3)/4	
PARAMETER I1MBUF = (PXBINS-1) + (3548+3)/4 + (18+3)/4	
INTEGER MSABUF(I1MBUF,NMBUFS)	ARRAY OF MSA PIXEL BUFFERS
INTEGER NBINSO(NMBUFS)	BIN NUMBER OF SAMPLE 0 FOR EACH MSA BUFFER
PARAMETER NDBUFS = 4	NUMBER OF DETECTION PIXEL BUFFERS IN ARRAY
INTEGERS IN DET BUF = INTS PREAMBLE + (NBINS+3)/4 + EXTRA INTS	
PARAMETER I1DBUF = (PXBINS-1) + (3548+2+3)/4 + 1	
INTEGER IDETBF(I1DBUF,NDBUFS)	ARRAY OF DETECTION PIXEL BUFFERS
INTEGER I1DREC	INTEGERS IN 1 DETECTION RECORD
INTEGER IDET(PXBINS)	GENERAL SHORT DETECTION BUFFER
INTEGER NOUTC	OUTPUT CHANNEL NUMBER (1 TO NOUTCH)
INTEGER NBUF	BUFFER NUMBER
INTEGER NDREC	DETECTION RECORD NUMBER
INTEGER MSALIN	CURRENT MSA LINE NUMBER
INTEGER MSALMN,MSALMX	MINIMUM, MAXIMUM MSA LINE IN OUTPUT WINDOW
INTEGER MSASAM	CURRENT MSA SAMPLE NUMBER
INTEGER MSASHN,MSASHX	MINIMUM, MAXIMUM MSA SAMPLE IN OUTPUT WINDOW
INTEGER MSASLO,MSASHI	LOW, HIGH DEFINED MSA SAMPLE IN CURRENT LINE
INTEGER IDEBIN,IDEBHI	CURRENT, HIGH DETECTION BIN NUMBER
PARAMETER NOINRA=254	RADIANCE DETECTION PIXEL 'NO INFO' FLAG
PARAMETER NODARA=255	RADIANCE DETECTION PIXEL 'NO DATA' THRESHOLD
INTEGER IOSTC	I/O STATUS CODE

```

C
C PROCEDURE
C -----
C
C      CALL TRACE
C
C
C INITIALIZE MINIMUM/MAXIMUM SCAN LINES/SAMPLES & DETECTION RECORD LENGTH
C
  IF(NOUTCH.GT.NDBUFS) CALL MDWARN(
  * 'MORE OUTPUT DETECTION CHANNELS THAN BUFFERS')
  IF(NOUTCH.GT.4) CALL MDWARN(
  * 'MORE OUTPUT DETECTION CHANNELS THAN PACKETS')
  IF(NDOTL.NE.0) GO TO 900
  MSALMN=MSADMM(WLIN.WMIN.NCCT)
  MSALNX=MSADMM(WLIN.WMAX.NCCT)
  MSASHN=MSADMM(WSAH.WMIN.NCCT)
  MSASHX=MSADMM(WSAH.WMAX.NCCT)
C
C      INTEGERS IN DET REC = #INTS PREAMBLE * (DBINS+3)/4
C      IIDREC      = (PXBINS-1) * (MSASHX-MSASHN+3+3)/4
C
  IF(IIDREC.GT.IIDBUF) CALL MDFATL( 'IIDREC > IIDBUF IN CLADE3')
  LDTRS(NCCT)=(IIDREC+27)/28 * CONVERT DETECT REC LENGTH TO UNIVAC SECTORS
C
C INITIALIZE CHANNEL SWITCHES
C
  NLINCH=4
  ASSIGN 320 TO LC2SMH
  IF(LINCH(2).LT.9) GO TO 130
  ASSIGN 360 TO LC2SMH
  NLINCH=1
  GO TO 180
130 ASSIGN 330 TO LC3SMH
  IF(LINCH(3).LT.9) GO TO 140
  ASSIGN 360 TO LC3SMH
  NLINCH=2
  GO TO 180
140 ASSIGN 340 TO LC4SMH
  IF(LINCH(4).LT.9) GO TO 150
  ASSIGN 360 TO LC4SMH
  NLINCH=3
150 CONTINUE
160 CONTINUE
C
C INITIALIZE PREAMBLE IN GENERAL SHORT DETECTION BUFFER
C
  IDET(PXRECH)=0
  IDET(PXLINO)=0
  IDET(PXCHAN)=1
  IDET(PXQUAL)=0
  IDET(PXBINT)='BYT'
  IDET(PXLBIN)=2
  IDET(PXLSAH)=0
  
```

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLADE3
004

```

      IDET(PXNBIN)=0
      IDET(PXNSAH)=0
      IDET(PXNODI)=NOINRA
      IDET(PXNDDA)=NODARA
      IDET(PXLJOI)=0
      IDET(PXNJOI)=0
C
C
C INITIALIZE BINS IN GENERAL SHORT DETECTION BUFFER TO 'NO INFO' FLAGS
C
      IDEBHI=NB4NI(1)
      DO 170 IDEBIN=1,IDEBHI
          CALL PUTBYT(IDET(PXBINS),(IDEBIN), NOINRA)
170 CONTINUE
C
C
C
C READ MSA. CLASSIFY. WRITE TO DETECTION FILE FROM LINE MSALNN TO MSALMX
C
      DO 700 MSALIN=MSALNN,MSALMX
C
C
C -- READ INPUT MSA DATA
C
      CALL GETRAD(MSABUF,(1)MBUF,(1)MDFUS,(1)OSTC,
      * MSALIN,MSASHN,MSASHX)
      * IF((1)OSTC.NE.'BADF').AND.
      & ((1)OSTC.NE.'OFL') GO TO 210
          CALL MDFATL(
      * CBS4CS((1)OSTC,1,4),' WHILE READING ON UNIT 3')
          GO TO 900
210 CONTINUE
C
C
C -- SECURITY BLANKET -- PRINT LINE NUMBER EVERY 90 LINES
C
      IF(MOD(MSALIN,90).NE.0) GO TO 220
          CALL MNOTE('LINE',CBS4IN(MSALIN,9))
          IF(MTRACE.NE.0) CALL PXDMP(MSABUF(1,1))
220 CONTINUE
C
C
C -- COMPUTE ACTUAL LO/HI SAMPLES READ AND BINS WITH SAMPLE 0
C
      MSASLO=-MAXINT
      MSASHI=-MAXINT
      DO 230 NBUF=1,NLIMCH
          NBINS(NBUF)=MSABUF(PXLBIN,NBUF)-MSABUF(PXLSAH,NBUF)
          MSASLO=MAX(MSASLO,MSABUF(PXLSAH,NBUF))
          MSASHI=MIN(MSASHI,MSABUF(PXNSAH,NBUF))
230 CONTINUE
C
C
C -- AFTER QUEUED I/O IS COMPLETE. CLEAR DET BUFFER(S) & INITIALIZE PREAMBLE(S)
C
      DO 280 NOUTC=1,NOUTCH
```

```

CALL WAITIO(IOSTC, LENPKY(I,NOUTC))
IF(IOSTC.NE.' ') CALL MDPATL(
  *   CS%CS(IOSTC,I.4), WHILE NR: NO DETECTION RECORD"
CALL MOVIST(IDETBF(I,NOUTC),(I),I,REC),
  *   IDET,(I),(PKBINS-I),IDET(PKBINS))
NOREC=(MSALIN-MSADMINLIN,MIN,NCCT)-NOUTCH-NOUTC
IDETBF(PKREC,NOUTC)=NOREC      8 RECORD NUMBER
IDETBF(PKLINE,NOUTC)=MSALIN    8 LINE NUMBER
IDETBF(PKCHAN,NOUTC)=NOUTC     8 OUTPUT CHANNEL NUMBER
IDETBF(PKQUAL,NOUTC)=MSABUF(PKQUAL,I)
IDETBF(PKLSAM,NOUTC)=MSASLO
IDETBF(PKMSAM,NOUTC)=MSASHI
IDETBF(PKMBIN,NOUTC)=IDETBF(PKLBIN,NOUTC)+MSASHI-MSASLO
200 CONTINUE
IF(INDTOTL.NE.0) GO TO 900

C
C
C -- SCREEN DATA IN BINS OF MSA BUFFER(S) TO FILL BINS OF DET BUFFER(S)
C
  IDLBIN=IDETBF(PKLBIN,I)-1
  DO 400 MSASAM=MSASLO,MSASHI
    IDEBIN=IDEBIN+1
    CALL GETBYT (INDVAL,
      *   MSABUF(PKBINS,I),(MSASAM+NBINS0(I)))
    IF((INDVAL.LT.0).OR.(INDVAL.GT.127)) GO TO 400
    GO TO LC2SMH 8 320 (MORE CHANNELS) OR 360 (LAST CHANNEL)
320  CALL GETBYT (MRAD2,
      *   MSABUF(PKBINS,2),(MSASAM+NBINS0(2)))
    IF(MRAD2.LT.LINVAL(INDVAL+1,1)) GO TO 400
    IF(MRAD2.GT.LINVAL(INDVAL+1,2)) GO TO 400
    GO TO LC3SMH 8 330 (MORE CHANNELS) OR 360 (LAST CHANNEL)
330  CALL GETBYT (MRAD3,
      *   MSABUF(PKBINS,3),(MSASAM+NBINS0(3)))
    IF(MRAD3.LT.LINVAL(INDVAL+1,3)) GO TO 400
    IF(MRAD3.GT.LINVAL(INDVAL+1,4)) GO TO 400
    GO TO LC4SMH 8 340 (MORE CHANNELS) OR 360 (LAST CHANNEL)
340  CALL GETBYT (MRAD4,
      *   MSABUF(PKBINS,4),(MSASAM+NBINS0(4)))
    IF(MRAD4.LT.LINVAL(INDVAL+1,5)) GO TO 400
    IF(MRAD4.GT.LINVAL(INDVAL+1,6)) GO TO 400
350  CONTINUE
360  CONTINUE
    CALL PUTBYT(IDETBF(PKBINS,I),(IDEBIN),
      *   MIND(INDVAL,127))
    IF(NOUTCH.EQ.1) GO TO 400
    CALL PUTBYT(IDETBF(PKBINS,2),(IDEBIN),
      *   MIND(MRAD2,127))
    IF(NOUTCH.EQ.2) GO TO 400
    CALL PUTBYT(IDETBF(PKBINS,3),(IDEBIN),
      *   MIND(MRAD3,127))
    IF(NOUTCH.EQ.3) GO TO 400
    CALL PUTBYT(IDETBF(PKBINS,4),(IDEBIN),
      *   MIND(MRAD4,127))
400 CONTINUE
C
C

```


DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLADE3
007

```
C      INTERNAL
      SUBROUTINE WAITIO( 8 WAIT FOR COMPLETION OF QUEUED I/O
0 IOSTC. 8 I/O STATUS CODE (SEE KOMIO FOR CODES)
      I IOPKT) 8 I/O PACKET FOR QUEUED I/O
C
C      INTEGER IOPKT(1)      8 ARGUMENT
C
C      CALL ERWAIT(IOPKT)
      IOSTC=IOCODE(IOPKT)
      RETURN
C
C      END
```


C THE SUBROUTINE.

C
 C
 C
 C
 C
 C

GLOBAL DECLARATIONS

```

INCLUDE KOMXQT.LIST      % COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST     % COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST     % COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST     % COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE WINDEF.LIST     % DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMIWH.LIST     % COMMON INPUT WINDOW PACKETS
INCLUDE KOMOHW.LIST     % COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMDET.LIST     % COMMON DETECTION FILE WINDOWS
INCLUDE KOML2N.LIST     % COMMON I/O PACKETS FOR DETECTION FILES (21-24)
INCLUDE KOMIO.LIST      % COMMON I/O FUNCTIONS
INCLUDE KOMSLM.LIST     % COMMON CLASSIFICATION SPECTRAL LIMITS
INCLUDE PXBDEF.LIST     % DEFINE PIXEL BUFFER STRUCTURE
INCLUDE MAXINT.LIST     % DEFINE MAXIMUM INTEGER
  
```

C
 C
 C
 C
 C

LOCAL DECLARATIONS

```

PARAMETER NMBUFS = 6      % # OF MSA PIXEL BUFFERS IN ARRAY
INTEGERS IN MSA BUF = #INTS PREAMBLE + (#BINS+3)/4 + (#EXTRA BYTES+3)/4
PARAMETER I1MBUF = (PXBINS-1) + (3548+3)/4 + (19+3)/4
INTEGER MSABUF(I1MBUF,NMBUFS) % ARRAY OF MSA PIXEL BUFFERS
INTEGER NBINSO(NMBUFS)    % BIN # OF SAMPLE 0 FOR EACH MSA BUFFER

PARAMETER NOBUFS = 4      % # OF DETECTION PIXEL BUFFERS IN ARRAY
INTEGERS IN DET BUF = #INTS PREAMBLE + (#BINS+5)/6 + #EXTRA INTS
PARAMETER I1DBUF = (PXBINS-1) + (3548+2+5)/6 + 1
INTEGER I1DET(I1DBUF,NOBUFS) % ARRAY OF DETECTION PIXEL BUFFERS
INTEGER I1DREC           % INTEGERS IN 1 DETECTION RECORD

INTEGER I1DET(PXBINS)    % GENERAL SHORT DETECTION BUFFER

INTEGER NBUF             % BUFFER NUMBER
INTEGER NDREC           % DETECTION RECORD NUMBER

INTEGER MSALIN           % CURRENT MSA LINE NUMBER
INTEGER MSALMN,MSALMX   % MINIMUM, MAXIMUM MSA LINE IN OUTPUT WINDOW
INTEGER MSASAM           % CURRENT MSA SAMPLE NUMBER
INTEGER MSASHN,MSASHX   % MINIMUM, MAXIMUM MSA SAMPLE IN OUTPUT WINDOW
INTEGER MSASLO,MSASHI   % LOW, HIGH DEFINED MSA SAMPLE IN CURRENT LINE

INTEGER I1DEBIN,I1DEBHI % CURRENT, HIGH DETECTION BIN NUMBER
INTEGER LOCDB5          % LOC IN DETECTION BUFFER OF CURRENT DETECTION BIN + 5

PARAMETER NOINDE=00     % DENSITY DETECTION PIXEL 'NO INFO' FLAG
PARAMETER NODADE=20     % DENSITY DETECTION PIXEL 'NO DATA' THRESHOLD

INTEGER INSTAT          % GETRAD INPUT STATUS CODE
  
```

C
 C

ROMAN NUMERALS: (I=1, X=10)

```

C
  INTEGER 100000 /*'100000'/
  INTEGER 110000 /*'110000'/ . X10000 /*'E10000'/
  INTEGER 111000 /*'111000'/ . 1X1000 /*'E1000'/
  INTEGER 011100 /*'011100'/ . 01X100 /*'0E100'/
  INTEGER 001110 /*'001110'/ . 001X10 /*'00E10'/
  INTEGER 000111 /*'000111'/ . 0001X1 /*'000E1'/
  INTEGER 000011 /*'000011'/ . 00001X /*'0000E'/
  INTEGER 000001 /*'000001'/

C
  INTEGER 010000 /*'010000'/
  INTEGER 001000 /*'001000'/
  INTEGER 000100 /*'000100'/
  INTEGER 000010 /*'000010'/

C
C
C PROCEDURE
C -----
C
  CALL TRACE

C
C
C INITIALIZE MINIMUM/MAXIMUM SCAN LINES/SAMPLES & DETECTION RECORD LENGTH
C
  MSALMN=MSADWH(NLIN,WMIN,NCCT)
  MSALMX=MSADWH(NLIN,WMAX,NCCT)
  MSASMN=MSADWH(WSAM,WMIN,NCCT)
  MSASMX=MSADWH(WSAM,WMAX,NCCT)
  MSALIN=MSALMN-1

C
C
C INTEGERS IN DET REC = #INTS PREAMBLE * (#BINS+5)/6
  I1DREC = (PXBINS-1) * (MSASMX-MSASMN+3+5)/6

C
  IF(I1DREC.GT.I1DBUF) CALL M0FATL( 'I1DREC > I1DBUF IN CLADE4')
  LDTRS(NCCT)=(I1DREC+27)/20 * CONVERT DETECT REC LENGTH TO UNIVAC SECTORS

C
C
C INITIALIZE CHANNEL SWITCHES
C
  NLIMCH=4
  ASSIGN 320 TO LC2SWH
  IF(LIMCH(2).LT.9) GO TO 130
  ASSIGN 360 TO LC2SWH
  NLIMCH=1
  GO TO 160
130 ASSIGN 330 TO LC3SWH
  ASSIGN 430 TO LC3SWH
  IF(LIMCH(3).LT.9) GO TO 140
  ASSIGN 360 TO LC3SWH
  ASSIGN 460 TO LC3SWH
  NLIMCH=2
  GO TO 160
140 ASSIGN 340 TO LC4SWH
  ASSIGN 440 TO LC4SWH
  IF(LIMCH(4).LT.9) GO TO 150
  ASSIGN 360 TO LC4SWH
  
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLADE4
004

```
                ASSIGN 480 TO LC4SMN
                NLINCH=3
150 CONTINUE
160 CONTINUE
C
C
C INITIALIZE PACKET AND BUFFER ASSIGNMENTS FOR DETECTION FILE
C
                CALL PKTDET(L2NPKT(1,NCCT))
                IDOUT=1
                IDIN2=2
                IDIN3=3
                IDIN4=4
                NDREC=0
C
C
C INITIALIZE PREAMBLE IN GENERAL SHORT DETECTION BUFFER
C
                IDET(PXRECN)=0
                IDET(PXLINO)=0
                IDET(PXCHAN)=1
                IDET(PXQUAL)=0
                IDET(PXBINT)='CHR'
                IDET(PXLBIN)=2 & BIN * IDET(PXLBIN)-1 & IDET(PXHBIN)+1 USED TO DERIVE JOIN
                IDET(PXLSAM)=0
                IDET(PXHBIN)=0
                IDET(PXHSAM)=0
                IDET(PXNOIN)=NOINDE
                IDET(PXNODA)=NODADE
                IDET(PXLJOI)=0
                IDET(PXHJOI)=0
C
C
C INITIALIZE BINS IN GENERAL SHORT DETECTION BUFFER TO 'NO INFO' FLAGS
C
                IDEBHI=NB*NI(1)
                DO 175 IDEBIN=1,IDEBHI
                    CALL PUTICE(IDET(PXBINS),(IDEBIN), NOINDE)
170 CONTINUE
C
C
C CLEAR ALL DETECTION BUFFERS
C
                DO 180 NBUF=1,NBUFS
                    CALL MOVIST(IDETBF(1,NBUF),(1),(11DREC),
                                IDET,(1),(PXBINS-1),IDET(PXBINS))
180 CONTINUE
C
C
C READ MSA DATA
C
                200 IF(MSALIN.OE.MSALMX) GO TO 500
                MSALIN=MSALIN+1
                CALL GETRAD(MSABUF,(11MBUF),(11NBUFS),INSTAT,
                            MSALIN,MSASMN,MSASMX)
                IF((INSTAT.NE.'BADF').AND.
```

```

      6 (INSTAT.NE.'OFL') GO TO 210
        CALL MCFATL
        CBS4CS(INSTAT.1.4) WHILE READING ON UNIT 3
        GO TO 900
C
C
C SECURITY BLANKET
C
210 IF(MOD(MSALIN.50).EQ.0) CALL MONOTE(
      - 'LINE'.CBS4IN(MSALIN.5))
C S IF(MOD(MSALIN.50).EQ.0) CALL PXBDMP(MSABUF(1.1)) & ***** TNP DEBUG *****
C
C
C UPDATE POINTERS & PREAMBLE OF OUTPUT BUFFER
C
      IDETBF(PXLINO.1BDIN3)=MSALIN      & LINE NUMBER
      IDETBF(PXQUAL.1BDIN3)=MSABUF(PXQUAL.1)
      MSASLO=-MAXINT
      MSASHI=+MAXINT
      DO 230 NBUF=1,NLIMCH
        NBINS0(NBUF)=MSABUF(PXLBIN.NBUF)-MSABUF(PXLSAM.NBUF)
        MSASLO=MAX0(MSASLO,MSABUF(PXLSAM.NBUF))
        MSASHI=MIN0(MSASHI,MSABUF(PXHSAM.NBUF))
230 CONTINUE
      IDETBF(PXLSAM.1BDIN3)=MSASLO
      IDETBF(PXHSAM.1BDIN3)=MSASHI
      IDETBF(PXHBIN.1BDIN3)=IDETBF(PXLBIN.1BDIN3)+MSASHI-MSASLO
C
C
C SCREEN DATA IN BINS OF MSA BUFFER TO FILL BINS OF DETECTION BUFFER
C
      IDEBIN=IDETBF(PXLBIN.1BDIN3)-1
      LOCDB5 = NC4NI(PXBINS-1) & * NUMBER OF CHARACTERS IN BUFFER PREAMBLE
      & * IDEBIN      & * CURRENT BIN NUMBER
      & * 5          & * 5 (50 DIVISION BY 6 YIELDS WORD NUMBER)
      DO 480 MSASAM=MSASLO,MSASHI
        LOCDB5=LOCDB5+1
        CALL GETBYT (INDVAL.
      - MSABUF(PXBINS.1).(MSASAM+NBINS0(1)))
C
C
C CLASSIFY PIXEL HI.
C
      IF((INDVAL.LT.0).OR.(INDVAL.GT.127)) GO TO 480
      GO TO LC2SHM & 320 (MORE CHANNELS) OR 360 (LAST CHANNEL)
320 CALL GETBYT (MRAD2.
      - MSABUF(PXBINS.2).(MSASAM+NBINS0(2)))
      IF(MRAD2.LT.LIMVAL(INDVAL+1.1)) GO TO 421
      IF(MRAD2.GT.LIMVAL(INDVAL+1.2)) GO TO 422
      GO TO LC3SHM & 330 (MORE CHANNELS) OR 360 (LAST CHANNEL)
330 CALL GETBYT (MRAD3.
      - MSABUF(PXBINS.3).(MSASAM+NBINS0(3)))
      IF(MRAD3.LT.LIMVAL(INDVAL+1.3)) GO TO 431
      IF(MRAD3.GT.LIMVAL(INDVAL+1.4)) GO TO 432
      GO TO LC4SHM & 340 (MORE CHANNELS) OR 360 (LAST CHANNEL)
340 CALL GETBYT (MRAD4.

```

```

      MSABUF(PXBINS.4).(MSASAM+NBINS0(4))
      IF(MRAD4.LT.LIMVAL(INOVAL+1.6)) GO TO 441
      IF(MRAD4.GT.LIMVAL(INOVAL+1.6)) GO TO 442
350      CONTINUE
360      CONTINUE
C
C
C ENUMERATE CLASSIFICATION DENSITY FOR HITS
C
370      NWD=LOCDB5/6
      NCHAR=MOD(LOCDB5,6)+1
      GO TO(371,372,373,374,375,376).NCHAR
371      IDETBF(NWD-1,18DIN4)=IDETBF(NWD-1,18DIN4)+000001
      IDETBF(NWD-1,18DIN3)=IDETBF(NWD-1,18DIN3)+000001
      IDETBF(NWD-1,18DIN2)=IDETBF(NWD-1,18DIN2)+000001
      IDETBF(NWD,18DIN4)  =IDETBF(NWD,18DIN4)  + 110000
      IDETBF(NWD,18DIN3)  =IDETBF(NWD,18DIN3)  + X10000
      IDETBF(NWD,18DIN2)  =IDETBF(NWD,18DIN2)  + 110000
      GO TO 480
372      IDETBF(NWD,18DIN4)  =IDETBF(NWD,18DIN4)  + 111000
      IDETBF(NWD,18DIN3)  =IDETBF(NWD,18DIN3)  + 1X1000
      IDETBF(NWD,18DIN2)  =IDETBF(NWD,18DIN2)  + 111000
      GO TO 480
373      IDETBF(NWD,18DIN4)  =IDETBF(NWD,18DIN4)  + 011100
      IDETBF(NWD,18DIN3)  =IDETBF(NWD,18DIN3)  + 01X100
      IDETBF(NWD,18DIN2)  =IDETBF(NWD,18DIN2)  + 011100
      GO TO 480
374      IDETBF(NWD,18DIN4)  =IDETBF(NWD,18DIN4)  + 001110
      IDETBF(NWD,18DIN3)  =IDETBF(NWD,18DIN3)  + 001X10
      IDETBF(NWD,18DIN2)  =IDETBF(NWD,18DIN2)  + 001110
      GO TO 480
375      IDETBF(NWD,18DIN4)  =IDETBF(NWD,18DIN4)  + 000111
      IDETBF(NWD,18DIN3)  =IDETBF(NWD,18DIN3)  + 0001X1
      IDETBF(NWD,18DIN2)  =IDETBF(NWD,18DIN2)  + 000111
      GO TO 480
376      IDETBF(NWD,18DIN4)  =IDETBF(NWD,18DIN4)  + 000011
      IDETBF(NWD,18DIN3)  =IDETBF(NWD,18DIN3)  + 00001X
      IDETBF(NWD,18DIN2)  =IDETBF(NWD,18DIN2)  + 000011
      IDETBF(NWD+1,18DIN4)=IDETBF(NWD+1,18DIN4)+ 100000
      IDETBF(NWD+1,18DIN3)=IDETBF(NWD+1,18DIN3)+ 100000
      IDETBF(NWD+1,18DIN2)=IDETBF(NWD+1,18DIN2)+ 100000
      GO TO 480
C
C
C CLASSIFY PIXEL NEAR HIT
C
421      IF(MRAD2.LT.LIMVAL(INOVAL+1.1)-LCVTOL) GO TO 480
422      IF(MRAD2.GT.LIMVAL(INOVAL+1.2)+LCVTOL) GO TO 480
      GO TO LC3SWN  430 (MORE CHANNELS) OR 480 (LAST CHANNEL)
430      CALL GETBYT (MRAD3,
      MSABUF(PXBINS.3).(MSASAM+NBINS0(3))
431      IF(MRAD3.LT.LIMVAL(INOVAL+1.3)-LCVTOL) GO TO 480
432      IF(MRAD3.GT.LIMVAL(INOVAL+1.4)+LCVTOL) GO TO 480
      GO TO LC4SWN  440 (MORE CHANNELS) OR 480 (LAST CHANNEL)
440      CALL GETBYT (MRAD4,
      MSABUF(PXBINS.4).(MSASAM+NBINS0(4))

```

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLADE4
007

```
441      IF(MRAD4.LT.LINVAL(INVAL+1.5)-LCVTOL) GO TO 480
442      IF(MRAD4.GT.LINVAL(INVAL+1.5)+LCVTOL) GO TO 480
450      CONTINUE
460      CONTINUE
```

C
C
C

C ENUMERATE CLASSIFICATION DENSITY FOR NEAR HITS

C

```
470      NWD=LOCDBS/8
          NCHAR=MOD(LOCDBS,8)+1
          GO TO(471,472,473,474,475,476), NCHAR
471      IDETBF(NWD,IBDIN3) = IDETBF(NWD,IBDIN3) + 100000
          GO TO 480
472      IDETBF(NWD,IBDIN3) = IDETBF(NWD,IBDIN3) + 010000
          GO TO 480
473      IDETBF(NWD,IBDIN3) = IDETBF(NWD,IBDIN3) + 001000
          GO TO 480
474      IDETBF(NWD,IBDIN3) = IDETBF(NWD,IBDIN3) + 000100
          GO TO 480
475      IDETBF(NWD,IBDIN3) = IDETBF(NWD,IBDIN3) + 000010
          GO TO 480
476      IDETBF(NWD,IBDIN3) = IDETBF(NWD,IBDIN3) + 000001
```

C
C
C

C LOOP BACK TO CLASSIFY NEXT PIXEL IN THIS LINE

C

```
480 CONTINUE
```

C

C

C CLEAR OUTPUT BUFFER & ROTATE DETECTION BUFFERS AFTER QUEUED I/O IS COMPLETE

C

```
500 CALL ERWAIT(L2NPKT(1,NCCT))
      CALL MOVIST(IDETBF(1,IBDOUT),(1),(1IDREC),
          IDET(1),(PXBINS-1),IDET(PXBINS))
      IBOTMP=IBDOUT
      IBDOUT=IBDIN2
      IBDIN2=IBDIN3
      IBDIN3=IBDIN4
      IBDIN4=IBOTMP
```

C
C
C
C

C QUEUE DETECTION RECORD OUTPUT BUFFER FOR I/O

C

```
      IF(IDETBF(PXLINO,IBDOUT).NE.0)
& CALL RITDBF(IDETBF(1,IBDOUT))
      IF(IDETBF(PXLINO,IBDOUT).LT.MSALMX) GO TO 200
```

C
C
C
C

C CLASSIFICATION COMPLETE

C

```
      MSADHW(HLIN,MMAX,NCCT)=MSALIN
500 CALL NVIATO(CLA000,NULSUB) & PREVENT CONSECUTIVE CALLS TO CLADE4
      RETURN
```

C
C
C

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLADE4
000

```
C
C
C
C   INTERNAL
C   SUBROUTINE PKTDET( 8 INITIALIZE DETECTION FILE I/O PACKET
U (XDPKT) 8 DETECTION PIXEL FILE I/O PACKET
C
C   INTEGER (XDPKT(1) 8 ARGUMENT
C   INTEGER (XDBUF(1) 8 ARGUMENT
C
C   IOSIZE((XDPKT)=11DREC
C   RETURN
C
C
C
C   ENTRY RITDBF( 8 WRITE RECORD FROM DETECTION BUFFER TO DETECTION FILE
U (XDBUF) 8 DETECTION PIXEL BUFFER
C
C   IF(ICODE((XDPKT).EQ.' ') GO TO 300
C   CALL MCFATL(
C   CBS4CS(ICODE((XDPKT).1.4).' WHILE WRITING DETECTION RECORD')
C   GO TO 900
300 NDREC=NDREC+1
C   (XDBUF(PXREC)=NDREC
C   CALL GETICE((XDBUF(PXLJ0)).
C   (XDBUF(PXBINS)).((XDBUF(PXLBIN)-1))
C   CALL GETICE((XDBUF(PXHJ0)).
C   (XDBUF(PXBINS)).((XDBUF(PXHBIN)-1))
C   IOSECT((XDPKT)=10+(NDREC-1)*LDTRS(NCCT) 8 NDREC + 1 GOES IN SECTOR 10
C   IOADDR((XDPKT)=LOC((XDBUF)
C   IOFUNC((XDPKT)='8C' 8 WRITE
C 8 IF(MOD((XDBUF(PXLINO).50).EQ.0) CALL PXBOMP((XDBUF) 8 ***** TMP DEBUG *****
C   CALL ERIO((XDPKT)
900 RETURN
C
C
C   END
```

```
      SUBROUTINE CLADE9  3 DETECT CLASS (PHASE 9)  
      -----  
C  
C  
C (E H SCHLOSSER)  
C  
C  
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED  
C -----  
C  
C      HDNOTE  
C      NVIATO  
C  
C  
C      EXTERNAL CLA129,CLADE9  
C      CALL TRACE  
C  
C  
C  
C  
C      CALL HDNOTE('DETECT.CLASS NOT YET IMPLEMENTED')  
C      CALL NVIATO(CLA129,PICDE9)  
C      RETURN  
C      END
```



```
      SUBROUTINE CLADE9 8 DETECT NSS-DERIVED DATA (PHASE 9)
      -----
C
C (E H SCHLOSSER)
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
C -----
C
C      MONOTE
C      MDCLRW
C
C
C      INCLUDE KONXQT.LIST
C      EXTERNAL CLA000.NULSUB
C      CALL TRACE
C
C
C ON RETURN, CALL CLA000 TO GET COMMANDS
C
C      CALL NVIATO(CLA000.NULSUB)
C
C
C
C CHECK DIAGNOSTIC COUNTERS
C
C      IF(NDWARN.EQ.0) GO TO 020
C      CALL MONOTE('PREVIOUS WARNINGS -- NO DETECTION FILE GENERATED')
C      IF(NDATCH.EQ.0) WRITE(6,015)
C      015 FORMAT(' ...TRY AGAIN')
C      GO TO 900
C      020 IF(NDFATL.EQ.0) GO TO 050
C      CALL MONOTE('PREVIOUS FATAL ERRORS - NO DETECTION FILE GENERATED')
C      GO TO 900
C      050 IF(MCHECK.EQ.0) GO TO 090
C      CALL MONOTE('CHECKOUT MODE -- NO DETECTION FILE GENERATED')
C      GO TO 900
C
C
C PREPARE FOR NEXT DISPLAY
C
C      090 NWNDOW=NWNDOW+1
C
C
C CLEAR WARNINGS 8 RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
C      900 CALL MDCLRW(-0)
C          RETURN
C          END
```

SUBROUTINE CLAEXI 8 TERMINATION ROUTINE FOR CLASSIFY

```
C
C
C HISTORY
C -----
C      E M SCHLOSSER      LEC      12/18/79      ORIGINAL CODE
C      E M SCHLOSSER      LEC      12/26/79      CALL CLSO2N. NOT CLOS2N & CLOS23
C
C METHOD
C -----
C      CLOSE INPUT & OUTPUT FILES & TERMINATE PROGRAM.
C
C MACHINE-DEPENDENT CODE
C -----
C      NONE.
C
C GLOBAL DECLARATIONS
C -----
C      INCLUDE KONXQT.LIST      8 COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
C      INCLUDE KONKLS.LIST      8 COMMON CLASSIFICATION INFO
C
C LOCAL DECLARATIONS
C -----
C      NONE.
C
C PROCEDURE
C -----
C      CALL TRACE
C
C      CLOSE/VERIFY INPUT MSA FILE (UNIT 3) IF OUTPUT DETECTION FILE WAS GENERATED
C
C      WRITE(6,125)
C      125 FORMAT('X,***PROGRAM TERMINATION')
C          IF(INDDATA.CE.0) 00 TO 900      8 DATA/CHECKOUT MODE
C              IF((KLSYTP.EQ.'RAD').OR.      8 IF OUTPUT FILE WAS GENERATED ...
C                  8 (KLSYTP.EQ.'DEN').OR.
C                  8 (KLSYTP.EQ.'CLA')) CALL CLOS23      8 ... THEN CLOSE INPUT FILE
C
C CLOSE OUTPUT FILES
C
C      IF(INMNDOM.07.0) CALL CLSO2N
C      CALL CLOS2R
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLAEKI
002

```
C
C
C TERMINATE PROGRAM
C
C 900 CALL PSTOP('***PLEASE BFREE 3. OR BREWIND 3. OR BLOCATE 3.')
```

```
C
C
C PSTOP DOES NOT RETURN
C
```

END

ROUTINE CLARAD: GET/CHECK RADIANCE LIMITS
(MID) : I: FIRST 3 CHARS OF COMMAND O: SPACES

(CLOSSER)

INTERNAL SUBROUTINES/FUNCTIONS CALLED

READS
NOLOG

INCLUDE KOMXQT.LIST : COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST : COMMON CLASSIFICATION INFO
INCLUDE KOMSLM.LIST : COMMON SPECTRAL LIMITS
CALL TRACE

CHECK POINTERS

IF(NLIMCH.GE.2) GO TO 220
 CALL MDHARN('NOT ENOUGH CHANNELS')
 GO TO 900
220 IF(NLIMCH.LE.4) GO TO 240
 CALL MDHARN('TOO MANY CHANNELS')
 GO TO 900
240 CONTINUE

C
C
C CHECK INDEX RADIANCE VALUES TO SEE IF CHANNELS HAVE JUST BEEN CHANGED
C
C IF(LCVLOI.LE.LCVHII) GO TO 500 : OLD CHANNELS -- UPDATE RADIANCE VALUES

C
C
C NEW CHANNELS -- INITIALIZE LIMIT RADIANCE VALUES
C

NLIMVA=(NLIMCH-1)*2
DO 450 INDVAL=0.127
 DO 440 N=1,NLIMVA,2
 LIMVAL(INDVAL+1,N) =+99999999 : MINIMUM
 LIMVAL(INDVAL+1,N+1)=-99999999 : MAXIMUM
440 CONTINUE
450 CONTINUE

C
C
C GET/CHECK INDEX RADIANCE VALUES
C

500 CALL GETSIN(INDVIN, 0.127, '*BAD MINIMUM RADIANCE --')
 CALL GETSIN(INDVAX, INDVIN, 127, '*BAD MAXIMUM RADIANCE --')
 LCVLOI=MIN0(LCVLOI,INDVIN)
 LCVHII=MAX0(LCVHII,INDVAX)
 INDVAL=INDVIN

C

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLARAD
002

```
C
C GET/CHECK LIMIT RADIANCE VALUES
C
  DO 550 N=2,NLIMCH
    IF(LIMCH(N).EQ.999) GO TO 550
    CALL GETSIN(LIMVAL(INDVAL+1.2*N-3),
      0.255,'BAD MIN RADIANCE --')
    CALL GETSIN(LIMVAL(INDVAL+1.2*N-2),
      0.255,'BAD MAX RADIANCE --')
    LCVLO(N)=MIN(LCVLO(N),LIMVAL(INDVAL+1.2*N-3))
    LCVHI(N)=MAX(LCVHI(N),LIMVAL(INDVAL+1.2*N-2))
  550 CONTINUE
    CALL GETSIN(ITEMP, +1.-1,'EXTRA RADIANCE SPECIFICATION --')
C
C
C DUPLICATE SAME LIMIT VALUES FOR REMAINING INDEX VALUES
C
  IF(INDVIN.EQ.INDVAX) GO TO 900
  INDVIN=INDVIN+1
  NLIM=(NLIMCH-1)*2
  DO 660 INDVAL=INDVIN,INDVAX
    DO 630 N=1,NLIM
      LIMVAL(INDVAL+1,N)=LIMVAL(INDVIN,N)
  630 CONTINUE
  660 CONTINUE
C
C
C DONE
C
  900 KOND=
  RETURN
C
  END
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLAROT
002

```
C LOCAL DECLARATIONS
C -----
C
C     INTEGER LOCFIL      & LOCATION WITHIN DISK SYMBOLIC FILE ( IF > 0 )
C
C PROCEDURE
C -----
C
C IDENTIFY PROGRAM
C
C     CALL PSTART( 'DAM CLASSIFY(8009)')
C
C OPEN ERTS FILE & LOAD REGISTRATION PARAMETERS
C
C     CALL OPEN3
C     CALL LDREG8      & FATAL ERROR IF REGISTRATION PARAMETERS NOT ON UNIT 8
C
C IDENTIFY ERTS SCENE
C
C     IF(MDATAC.NE.0) GO TO 300      & DATA/CHECKOUT MODE
C     IF(NDFATL.EQ.0)
C       & ENCODE(10,195,JHD0) NERTS.NCCT.NCCTOT
C 185 FORMAT(' E-.J1.J4.'-' .J5.IX.JI.'/' .JI) & COMPACT TO FIT ON BOX PAGE
C     WRITE(6,185) NERTS.NCCT.NCCTOT
C     WRITE(6,195)
C 195 FORMAT(1X)      & SKIP LINE
C     CALL IOLU3( 6)
C     CALL IDERTS( 6)
C
C
C INITIALIZE SPACING
C
C     MSAOWH(WLIN.WSP100)=100
C     MSAOWH(WSAM.WSP100)=100
C
C
C QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE
C
C 300 CALL SYSADD(LOCFIL, 'MACDAM','DEF-CLASSIFY',. .)
C     IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL, 'DAM','DEF-CLASSIFY',. .)
C     IF(LOCFIL.LE.0) CALL MCFATL( 'NO DEFAULT COMMANDS')
C
C
C RETURN. THEN HAVE MAIN CALL CL4000 TO GET DEFAULT/USER COMMANDS
C
C     CALL NVIATO( CL4000.NULSUB)
C     RETURN
C     END
```



```

      INCLUDE FACBIT.LIST      & EXEC-8 FACILITY REQUEST STATUS BIT MNEMONICS
C
C
C LOCAL DECLARATIONS
C -----
C
      INTEGER NWD              & WORD NUMBER IN HEADER BUFFER AND/OR COMMON BLOCK
      INTEGER KBUFR(20,9)     & INTERNAL WORK BUFFER FOR DETECTION FILE HEADERS
C
      INTEGER
C CHAR      00000000011111111122222222223
C           123456789012345678901234567890
      & JASO2(4)/'&ASO,AX '&AMDET-N. . . . . '/'
      & JFREE2(4)/'&FREE      2N. . . . . '/'
      & JFRED2(4)/'&FREE.D    2N. . . . . '/'
C
C
C PROCEDURE
C -----
C
      CALL TRACE
C
C
C IS DETECTION FILE NON-EXISTENT, GOOD-OLD, BAD, GOOD-NEW?
C
      IF( (NCCT.LT.1).OR.(NCCT.GT.4) ) GO TO 900
      IF(KLSTYP.EQ.'NUL') GO TO 800      & FILE DOES NOT (OR SHOULD NOT) EXIST
      IF(KLSTYP.EQ.'OLD') GO TO 900     & DON'T TOUCH THAT GOOD OL' FILE!!
      IF(INDFATL.NE.0) GO TO 900       & FILE BAD -- DON'T REPLACE INTERIM HEADER
C
C
C GOOD NEW FILE -- MOVE KOMNER, KOMKLS, KOMFIT, AND KOMDET INTO BUFFER
C
      DO 811 NWD=1,SIZNER
      811 KBUFR(NWD,1)=KOMNER(NWD)
      DO 812 NWD=1,SIZKLS
      812 KBUFR(NWD,4)=KOMKLS(NWD)
      DO 813 NWD=1,SIZFIT
      813 KBUFR(NWD,8)=KOMFIT(NWD)
      DO 815 NWD=1,SIZDET
      815 KBUFR(NWD,8)=KOMDET(NWD)
C
C
C WRITE FINAL HEADERS FROM BUFFER
C
      CALL WRHDR(L2NPKT(1,NCCT))
C
C
C CATALOG DETECTION FILE
C
      800 CALL CST4IN(JFREE2,17.1, NCCT,1)
      CALL ERCSF(NAO,JFREE2)
      CALL CST4IN(JASO2,17.1, NCCT,1)
      CALL ERCSF(NAO,JASO2) & PROTECT UNTIL RUN TERMINATES
      IF(KLSTYP.EQ.'NUL') CALL CST4IN(JFRED2,17.1, NCCT,1)
      IF(KLSTYP.EQ.'NUL') CALL ERCSF(NAO,JFRED2)
  
```


SUBROUTINE OPNO2N : OPEN/CHECK OUTPUT DETECTION FILE

C
C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 01/03/74 ORIGINAL CODE
C MARY TOMPKINS LEC 12/13/79 REV HEADER & REPLACE FLD BY CST4IN
C MARY TOMPKINS LENSCO 09/16/80 PUT INTERN DET FILE NAME IN I/O PKTS
C
C
C METHOD
C -----
C
C ATTEMPT TO ASSIGN NEW DETECTION FILE. IF FILE IS ALREADY ASSIGNED
C OR FILE IS CATALOGED AND NOT PRESENTLY IN USE READ HEADER OF
C FILE INTO BUFFER. COMPARE CONTENTS OF BUFFER TO KOMNER,
C KOMKLS,KOMFIT,AND KOMDET. IF COMPARASION IS NOT EQUAL REINITIALIZE
C DETECTION FILE HEADER TO BINARY ZEROS BEFORE RETURNING.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C ONE UNIVAC FASTRAND-FORMATTED DISK SECTOR IS 28 UNIVAC 1100 WORDS.
C
C
C EXTERNAL REFERENCES
C -----
C
C ERCSF : SUBMIT EXEC-8 CONTROL STATEMENT FUNCTION
C MNOTE : PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
C MDFATL : PRINT/COUNT/LOG 'FATAL ERROR' DIAGNOSTIC MESSAGE
C ER10W : INITIATE I/O & WAIT FOR COMPLETION
C CST4IN : CHAR STRING FOR INTEGER
C DOUBLE PRECISION CBS4CS : VARIABLE LENGTH CST FOR FIXED LENGTH CST
C
C
C EXCEPTIONS
C -----
C
C 1. IF THE DETECTION FILE EXISTS BUT IS BEING USED BY ANOTHER RUN.
C THEN ISSUE 'FATAL ERROR' AND RETURN.
C
C 2. IF I/O ERRORS ARE ENCOUNTERED WHILE READING OLD HEADER OR WRITING
C INTERIM BLANK HEADER. THEN ISSUE 'FATAL ERROR' AND RETURN.
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST : COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C INCLUDE KOMI2N.LIST : COMMON I/O PACKETS FOR DETECTION FILES (21-24)
C INCLUDE KOMI0.LIST : COMMON I/O FUNCTION:
C INCLUDE KOMNER.LIST : COMMON ERTS SCENE PARAMETERS

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

OPROZ
002

```
INCLUDE KOMKLS.LIST      8 COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST     8 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMOHM.LIST    8 COMMON OUTPUT WINDOW PACKETS
INCLUDE KONDET.LIST    8 COMMON DETECTION FILE WINDOWS
INCLUDE WINDEF.LIST    8 DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE PKBDEF.LIST    8 DEFINE STRUCTURE OF PIXEL BUFFER
INCLUDE FACBIT.LIST    8 EXEC-8 FACILITY REQUEST STATUS BIT MNEMONICS
```

C
C
C LOCAL DECLARATIONS
C -----
C

```
      INTEGER
C CHAR      000000000111111111122222222223333333333444444444
           12345678901234567890123456789012345678901234567
C          7 JASOC2(8)/'SASO.CP' *DAMDET-N..F/1/POS/48 . DETECTION FILE '/'
C          4 JASOAZ(4)/'SASO.AX' *DAMDET-N. . . '/'
C          4 JUSEZ(4) /'SUSE 2N.' *DAMDET-N. . . '/'
```

C
 INTEGER KBUFR(20,9) 8 INTERNAL WORK BUFFER FOR DETECTION FILE HEADERS
 INTEGER IROCNT 8 COMPARISON ERROR COUNTER
 INTEGER ISTAT 8 I/O STATUS CODE (CST)

C
C
C PROCEDURE
C -----
C

```
      CALL TRACE
C  
C  
C INITIALIZE COMPARISON ERROR COUNTER  
C  
      IROCNT=0
```

C
C
C INITIALIZE DETECTION WINDOW, DATE, AND RECORD LENGTH IN SECTORS
C

```
      DO 110 NCT=1,4
           MSADWH(WLIN,WMIN,NCT)=0
           MSADWH(WLIN,WMAX,NCT)=0
           MSADWH(WSAM,WMIN,NCT)=0
           MSADWH(WSAM,WMAX,NCT)=0
           JENMDY(NCT)=' '
           LOETRS(NCT)=0
110 CONTINUE
      IF((NCCT.LT.1).OR.(NCCT.GT.4)) GO TO 900
           MSADWH(WLIN,WMIN,NCCT)=MSAOWH(WLIN,WMIN)
           MSADWH(WLIN,WMAX,NCCT)=MSAOWH(WLIN,WMAX)
           MSADWH(WSAM,WMIN,NCCT)=MSAOWH(WSAM,WMIN)
           MSADWH(WSAM,WMAX,NCCT)=MSAOWH(WSAM,WMAX)
           JENMDY(NCCT)=JMDY
           LOETRS(NCCT)=0      8 RECORD LENGTH IN SECTORS NOT DEFINED AT FILE OPEN
```

C
C
C PUT (20*NCCT) INTERNAL DETECTION FILE NAME IN ALL LUZN I/O PACKETS
C

```

    DO 120 NPKT=1,4
      CALL MOVCS(L2NPKT(1,NPKT),(1),(12),
        '2',(1),(1),' ')
      CALL PUTICE(L2NPKT(1,NPKT),(2),
        ICE('0')+NCCT)
120 CONTINUE
C
C
C TRY TO ASSIGN NEW DETECTION FILE
C
    CALL CST4IN(JASOC2,17.1, NCCT,1)
    WRITE(6,125) JASOC2
125 FORMAT('0(',BAG,'1')
    CALL ERCSF(NAO,JASOC2)
    CALL CST4IN(JUSE2,7.1, NCCT,1)
    CALL CST4IN(JUSE2,17.1, NCCT,1)
    CALL ERCSF(ITEMP,JUSE2)
    IF(ALREDY(NAO).EQ.0) GO TO 150
    CALL MONOTE('DETECTION FILE ALREADY ASSIGNED TO THIS RUN')
    GO TO 400
150 IF(PRYCAT(NAO).EQ.0) GO TO 900 & SUCCESSFULLY ASSIGNED NEW DETECTION FILE
C
C
C OLD DETECTION FILE EXISTS -- ASSIGN IT
C
    CALL MONOTE('DETECTION FILE ALREADY CATALOGED')
    CALL CST4IN(JASO2,17.1, NCCT,1)
    CALL ERCSF(NAO,JASO2)
    IF(OTHRUN(NAO).EQ.0) GO TO 400
    CALL MDFATL(
      'DETECTION FILE IN USE BY ANOTHER RUN - RUNID/QUAL NOT UNIQUE')
    GO TO 900
C
C
C READ HEADER (SECTORS 0 THRU 6) INTO BUFFER
C
400 CALL RDHDR
C
C
C COMPARE SECTORS 0 THRU 6 WITH KOMNER, KOMKLS, AND KOMFIT
C
    DO 441 NWD=1,62
      IF(KOMNER(NWD).NE.KBUFR(NWD,1)) IROCNT=IROCNT+1
441 CONTINUE
    DO 442 NWD=1,28
      IF(KOMKLS(NWD).NE.KBUFR(NWD,4)) IROCNT=IROCNT+1
442 CONTINUE
    DO 443 NWD=1,17
      IF(KOMFIT(NWD).NE.KBUFR(NWD,6)) IROCNT=IROCNT+1
443 CONTINUE
C
C
C COMPARE KOMDET WITH SECTOR 7 THRU 8
C
    DO 454 NWD=1,24 & DATES NEED NOT MATCH
      IF(KOMDET(NWD).NE.XBUFR(NWD,8)) IROCNT=IROCNT+1

```



```
      SUBROUTINE OPRCLA  & OPEN ALTERNATE PRINT FILE FOR CLASSIFY
      -----
C
C
C HISTORY
C -----
C      E H SCHLOSSER      LEC      06/20/78      ORIGINAL CODE
C
C
C METHOD
C -----
C      A MAXIMUM OF 1 ALTERNATE PRINT FILE IS OPENED & INITIALIZED.
C
C MACHINE-DEPENDENT CODE
C -----
C      NONE.
C
C EXTERNAL REFERENCES
C -----
C      OPENPR      OPEN ALTERNATE PRINT FILES
C
C EXCEPTIONS
C -----
C      NONE.
C
C      INCLUDE KOMXQT.LIST
C      CALL TRACE
C
C NO MORE THAN 1 ALTERNATE PRINT FILE FOR CLASSIFY
C
C      IF(MALTH.LE.0) GO TO 900      & NO ALTERNATE PRINT FILES
C      MALTH=MIND(MALTH.1)
C
C OPEN FILE(S)
C
C      IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE
C      CALL OPENPR
C
C IDENTIFY ERTS SCENE
C
C      WRITE(10,225)
C      225 FORMAT(1X)      & SKIP LINE
C      CALL IDLUS(10)
C      CALL IDERTS(10)
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

OPRCLA
002

C
C
000 RETURN
END

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

SLMCLA
002

```
360 CONTINUE
    IF(MOD(LCV,10).NE.0) GO TO 380
    DO 370 INDVAL=ICV10,ICVH,10
370 IF(LSYM(INDVAL+1).EQ.' ') LSYM(INDVAL+1)='+'
380 WRITE(10,385) LCV,(LSYM(INDVAL+1),INDVAL=ICVL,ICVH)
385 FORMAT(1X,J3,128A1)
390 CONTINUE
```

C

C

C WRITE GRAPHIC SCALE FOR INDEX CHANNEL

C

```
400 FORMAT(4X,128I1)
    DO 100 INDVAL=ICVL,ICVH
100 LSYM(INDVAL+1)=INDVAL/100
    WRITE(10,400) (LSYM(INDVAL+1),INDVAL=ICVL,ICVH)
    DO 10 INDVAL=ICVL,ICVH
10 LSYM(INDVAL+1)=(INDVAL-100*(INDVAL/100))/10
    WRITE(10,400) (LSYM(INDVAL+1),INDVAL=ICVL,ICVH)
    DO 1 INDVAL=ICVL,ICVH
1 LSYM(INDVAL+1)=INDVAL-10*(INDVAL/10)
    WRITE(10,400) (LSYM(INDVAL+1),INDVAL=ICVL,ICVH)
    WRITE(10,500) LIMCHI
500 FORMAT('0 CHANNEL ',11)
    RETURN
    END
```

C PROGRAM PRDDET

C -----

C

C

C HISTORY

C -----

C

C

E H SCHLOSSER

LEC

07/02/73

ORIGINAL CODE

C

E H SCHLOSSER

LEMSCO

05/16/80

KOMALT/SYM/TBL/KS.PRDENS TO PRDDET

C

C

C METHOD

C -----

C

C

THIS PROGRAM DISPLAYS AND/OR LISTS PORTIONS OF THE
DETECTION FILE(S) GENERATED BY THE CLASSIFY PROGRAM.

C

C

C

C MACHINE-DEPENDENT CODE

C -----

C

C

NONE.

C

C

C EXTERNAL REFERENCES

C -----

C

C

NVIATO & NAME 'VIA' 'TO' ROUTINES

C

VIATO & CALL 'VIA' 'TO' ROUTINES

C

VIA TO

EXTERNAL PRO000. PRDXQT

C

C

C EXCEPTIONS

C -----

C

C

1. IF NO DETECTION FILES EXIST FOR THE CURRENT PROJECT
AND RUNID, OR THE EXISTING DETECTION FILE(S) ARE DEFECTIVE
OR INCOMPATIBLE THEN PRDDET WILL GENERATE A FATAL
ERROR AND REFUSE TO GENERATE ANY DISPLAYS OR LISTS.

C

C

C

C

C

2. IF PRDXQT DOES NOT CALL NVIATO TO CHANGE THE 'VIA' AND/OR 'TO'
ROUTINES, THEN PRDDET WILL CALL TO PRDXQT IN AN ENDLESS LOOP!

C

C

C

C GLOBAL DECLARATIONS

C -----

C

INCLUDE KOMXQT.LIST

& COMMON PROGRAM EXECUTION SWITCHES, COUNTERS

INCLUDE KOMLOG.LIST

& COMMON LOG FILE BUFFER, I/O PKT, POINTERS

INCLUDE KOMLUS.LIST

& COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5

INCLUDE KOML2N.LIST

& COMMON I/O PKTS FOR DETECTION FILES (UNITS 2N

INCLUDE KOMIHW.LIST

& COMMON INPUT WINDOW PACKETS

INCLUDE KOMOHV.LIST

& COMMON OUTPUT WINDOW PACKETS

INCLUDE KOMNER.LIST

& COMMON ERTS SCENE PARAMETERS

INCLUDE KOMKLS.LIST

& COMMON CLASSIFICATION INFO

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

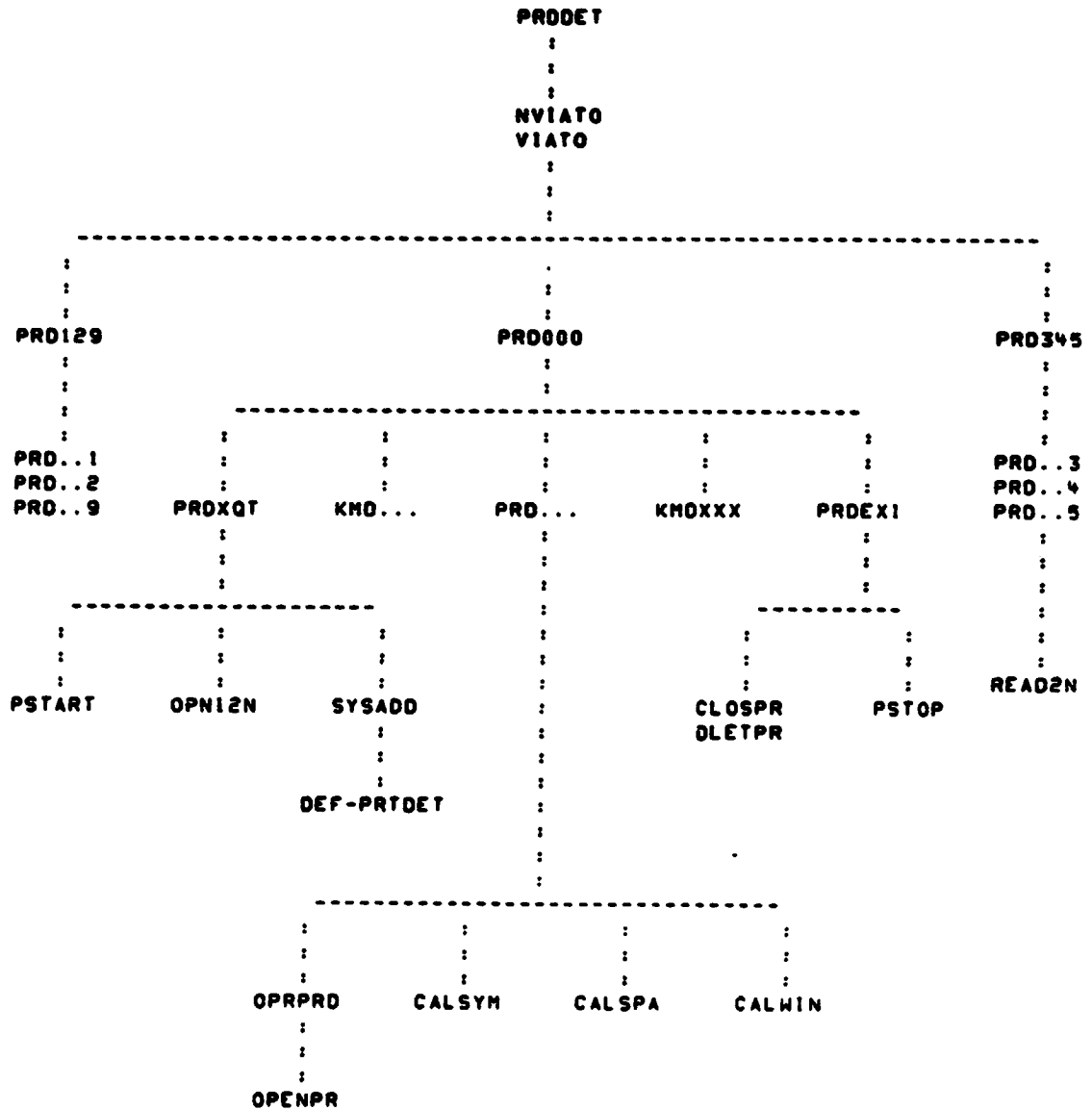
PRDET
002

INCLUDE KONFIT.LIST	8 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KONALT.LIST	8 COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS
INCLUDE KONSYN.LIST	8 COMMON SYMBOL TABLE
INCLUDE KONTBL.LIST	8 COMMON MULTI-PURPOSE TABLE
INCLUDE KONKS.LIST	8 COMMON COLOR SCREEN PARAMETERS
INCLUDE KONDET.LIST	8 COMMON DETECTION FILE WINDOW PACKETS & DATES

C
C
C PROCEDURE
C -----
C

```
      CALL NVIATO( PRD000,PRDXQT)      8 FIRST CALL IS VIA PRD000 TO PRDXQT
100 CONTINUE
      CALL VIATO
GO TO 100
END      8 (STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE)
```

PRTDET HIERARCHY



PROGRAM PRDDET/VIRTUAL

HISTORY

E H SCHLOSSER	LEC	08/02/74	ORIGINAL CODE
E H SCHLOSSER	LEC	11/06/79	MAP.FZ(N): NO 'N' IN DEMAND
E H SCHLOSSER	LEMSCO	05/16/80	CHANGE PRDENS TO PRDDET

METHOD

CONSTRUCT MAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT XQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE MAP & XQT COMMANDS TO TEMPORARY FILE 20.
ADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 8-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES.
DIFFERENT OPERATING SYSTEMS. AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS	FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IONS	INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS	TERMINATE PROGRAM EXECUTION
DAM.PRDET-MAP	SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAPOPT	STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS SASO-D & APREP-D.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

(2	REAL TIME
(3	LOW EXEC
(4	DEMAND
(5	DEADLINE BATCH
(6	BATCH

(BXQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER AS IN
 MASTER BIT NOTATION.)

LOCAL DECLARATIONS

```

      AXRS
S(00) . D-BANK
SSSH   FORM      6.6.6.18
      1111122222333333444445555556666667777788888999999
LABSDF  SSSH      050.1.'F'.0 . LABEL. 1 WD. FORTRAN. FIELDATA
LABINO  *SDF*
MAPSDF  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
MAPINO  *BXQTS: MAP.FZN DAM.PRDDET-MAP.PRDDET . :BXQTS*
ADDSDF  SSSH      000.9.0.0
ADDINO  *BXQTS: ADD DAM.SYS-MAPOPT . :BXQTS*
XQTSDF  SSSH      000.9.0.0
XQTING  *BXQTS: XQT.1 . PRDDET . :BXQTS*
E0FSDF  - 0 . END-OF-FILE STOP WORD
PF      FORM      12.6.18
CSFASO  *BASO.T 20. . .
CSFADD  *BADD 20. . .
SAVREG  RES      1
IOPKT   ISOD      '20'.WS 33.LABSDF.'0' 0
  
```

PROCEDURE

```

S(01) . I-BANK
PRDENS* LA,U      A0. . . . . A0 := .
      TNE.U      A4.4 . . . . . SKIP NEXT INST IF A4<>4 (NOT DEMAND)
      SA.S2      A0.MAPINO+2 . DEMAND! BLANK OUT N OPTION
      LA        A0.(CSFASO) . ADDRESS OF BASO IMAGE
      ER        CSFS . DO IT
      SA        A0.SAVREG . STORE &
      PSRINT    (PF 2.1.SAVREG) . PRINT BASO STATUS
      GETOPT    . LOAD OPT LTRS INTO A2.A3.A4
      PUTOPT    DS      A2.XQTING+2 . STORE OPTION LETTERS INTO BXQT IMAGE
      SA        A4.XQTING+4 . (3 WORDS -- MAX 18 OPT LETTERS)
      WRITE     LA      A0.(IOPKT) . ADDRESS OF I/O PACKET
      ER        IOWS . WRITE SDF IMAGES TO 20.
      ADD       LA      A0.(CSFADD) . ADDRESS OF BADD IMAGE
      ER        CSFS . DO IT
      ER        EXITS
      END       PRDENS
  
```

PRDDET OVERLAY STRUCTURE

HISTORY

E H SCHLOSSER	LEC	02/19/74	ORIGINAL CODE
E H SCHLOSSER	LEC	07/19/78	UPGRADE DOCUMENTATION
E H SCHLOSSER	LEC	01/30/79	ADD MACRO COMMANDS
E H SCHLOSSER	LEC	01/08/80	PEEK,POKE,IF,FI & PHASE 3,4,5 SEG
E H SCHLOSSER	LENSCO	05/16/80	CHA.COL.INT.PIC.TAB, PRDDEMS TO DET

LIB DAM.

SEG S-MAIN

IN DAM.PRDDET/ . MAIN PROGRAM
IN DAM.NVIATO . NAME/CALL 'VIA' AND 'TO' SUBROUTINES
IN DAM.NULSUB . DO NOTHING
IN DAM.SYS-BLOCK . BLOCK DATA SUBROUTINE

MONITOR FOR PHASE 0.1.2.9 COMMANDS -----

SEG S-PRD0129*

IN DAM.PR0000 . CALL USER-SPECIFIED PHASE 0 ROUTINE
IN DAM.PR0129 . CALL PREVIOUSLY NAMED PHASE 1/2/9 'TO' ROUTINE
IN DAM.NTABS/DAM . DAM UNIT # TABLE GOES IN SAME SEG W/ FORTRAN I/O

UTILITIES FOR PHASE 0.1.2.9 COMMANDS -----

SEG S-READS*.(S-PRD0129)

IN DAM.READS . 'READ' INTO UNIT 5 BUFFER
IN DAM.GET5 . 'GET' FREE-FORMAT FIELD FROM UNIT 5 BUFFER
IN DAM.WARNS . PROCESS WARNING DIAGNOSTIC FOR UNIT 5 FIELD
IN DAM.SPANS . ENABLE/DIABLE SPANNING FOR UNIT 5

SEG S-OPNCLPR*.(S-PRD0129)

IN DAM.OPRPRD . OPEN ALT PRT FILES
IN DAM.CLOSPR . CLOSE ALT PRT FILES

SEG S-CALSWIN*.(S-PRD0129)

IN DAM.CALSYM . CALIBRATE SYMBOL TABLE
IN DAM.CALSPA . CALIBRATE PRINT/PLOT COEFFICIENTS FOR SPACING
IN DAM.CALWIN . CALIBRATE WINDOW

PHASE 0.1.2.9 COMMANDS (FORTRAN I/O ALLOWED) -----

SEG S-XQTEXI*.(S-READS,S-OPNCLPR,S-CALSWIN)
IN DAM.PROXQT . PRDDET INITIALIZATION ROUTINE
IN DAM.PRDEXI . PRDDET TERMINATION ROUTINE

SEG S-PSTART*.(S-XQTEXI)

IN DAM.PSTART . GENERAL INITIALIZATION ROUTINE

SEG S-OPN12N*. (S-XQTEX1)

IN DAM.OPN12N . OPEN INPUT DETECTION FILE(S) (UNITS 21 ... 24)

SEG S-PSTOP*. (S-XQTEX1)

IN DAM.PSTOP . GENERAL TERMINATION ROUTINE

SEG S-HELP*. (S-READS.S-OPNCLPR.S-CALSWIN)

IN DAM.KMDCLE . CLEAR WARNINGS/ERRORS
IN DAM.KMDEXP . EXPLAIN PROGRAM/COMMAND
IN DAM.KMDFI . END IF...FI BLOCK
IN DAM.KMDIF . BEGIN IF...FI BLOCK
IN DAM.KMONEW . PRINT NEWS
IN DAM.KMDNEX . CONDITIONALLY PERFORM NEXT COMMAND
IN DAM.KMDOFF . TURN OFF MODE SWITCH(ES)
IN DAM.KMDON . TURN ON MODE SWITCH(ES)
IN DAM.KMDOPEE . PEEK
IN DAM.KMDOPOK . POKE
IN DAM.KMDTIM . PRINT CLOCK TIME & CHARGE TIME

SEG S-GEOMETRY*. (S-READS.S-OPNCLPR.S-CALSWIN)

IN DAM.KMOMER . GET/CHECK TRANSVERSE MERCATOR CENT MERIDIAN
IN DAM.KMOSPA . GET/CHECK WINDOW SPACING
IN DAM.KMOTIC . GET/CHECK TICK INTERVALS
IN DAM.KMDWIN . GET/CHECK WINDOW ENVELOPE/VERTICES
IN DAM.KMDZON . GET/CHECK UTM PROJECTION ZONE

SEG S-SPECS-9*. (S-READS.S-OPNCLPR.S-CALSWIN)

IN DAM.DETCHA . GET/CHECK DETECTION CHANNELS
IN DAM.KMDDEN . GET/CHECK DENSITY LIMITS
IN DAM.KMDHEA . GET/CHECK PAGE HEADING(S)
IN DAM.KMDORI . GET/CHECK WINDOW ORIGIN
IN DAM.KMDRAD . GET/CHECK RADIANCE LIMITS
IN DAM.PRODI9 . DISPLAY (PHASE 9)
IN DAM.PROLI9 . LIST (PHASE 9)
IN DAM.PROPI9 . PICTURE (PHASE 9)

SEG S-MISC*. (S-READS.S-OPNCLPR.S-CALSWIN)

IN DAM.KMDCOL . GET/CHECK COLORS
IN DAM.KMDCOP . GET/CHECK NUMBER OF OUTPUT COPIES
IN DAM.KMDINT . GET/CHECK INTENSITY
IN DAM.KMDPAG . SKIP TO TOP OF NEXT PAGE
IN DAM.KMDPRI . GET/CHECK PRINTER SPECIFICATIONS
IN DAM.KMDREN . RENUMBER (GET/CHECK NEW WINDOW SEQUENCE NUMBER)
IN DAM.KMDSYM . GET/CHECK SYMBOLS

SEG S-EXEC*. (S-READS.S-OPNCLPR.S-CALSWIN)

IN DAM.KM0XXX . MACRO COMMANDS
IN DAM.KM00AD . DYNAMIC 0ADD
IN DAM.KM00AS . DYNAMIC 0ASO
IN DAM.KM00BR . DYNAMIC 0BRKPT
IN DAM.KM00FR . DYNAMIC 0FREE
IN DAM.KM00LO . DYNAMIC 0LOG

SEG S-DISLISPIC*. (S-READS.S-OPNCLPR.S-CALSWIN)

IN DAM.PRODIS . DISPLAY (PHASE 0)
IN DAM.PROLIS . LIST (PHASE 0)
IN DAM.PROPIC . PICTURE (PHASE 0)

SEG S-HISTAB*. (S-READS.3-OPNCLPR.S-CALSWIN)
IN DAM.KMCRO . CROSSTABULATE COLOR & INTENSITY
IN DAM.KMOTAB . TABULATE PREVIOUSLY DIS/LIS/PIC DATA

. MONITOR FOR PHASE 3.4.5 COMMANDS -----

SEG S-PRD345*. S-PRD0129
IN DAM.PRO345 . CALL PREVIOUSLY NAMED PHASE 3/4/5 'TO' ROUTINE

. PHASE 3.4.5 COMMANDS (NO FORTRAN I/O) -----

SEG S-PRD13*. (S-PRD345) . DISPLAY DETECTION DATA
IN DAM.PROD13

SEG S-PROL13*. (S-PRD345) . LIST DETECTION DATA
IN DAM.PROL13

SEG S-PROPI3*. (S-PRD345) . PICTURE DETECTION DATA
IN DAM.PROPI3

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRTDET-NAP/VIRTUAL
001

IN DAM.PRTDET/VIRTUAL

INTEGER KOMD
INTEGER LSSTAT
INTEGER KASE

8 FIRST 3 CHARS OF USER COMMAND (BLANK AFTER DONE)
8 READS STATUS ('EOF' MEANS END-OF-FILE)
8 MODIFIED I-C-E OF FIRST CHAR OF COMMAND

PROCEDURE

```

C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
C CALL TRACE
C CALL NANSUB      8 CALL TO NULSUB DOES NOTHING
C
C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C
C KOMD=' NUL'      8 IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
C IF (NBATCH.NE.0) CALL READSILSSTAT.      8 FILL BUFFER, BLANK CUE MSO
C IF (NBATCH.EQ.0) CALL READSILSSTAT.      NULCST)      8 FILL BUFFER, NO CUE MSO
C IF (LSSTAT.NE.' ') KOMD='EOFS'
C IF (KOMD.NE.'EOFS') CALL GETSAL (KOMD.(3), NULCST)      8 GET 3 ALPHA CHARS
C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C
C KASE=ICE (KOMD)-ICE ('A')+1      8 A TO Z = 1 TO 26
C
C CASE STATEMENT ON MODIFIED I-C-E OF COMMAND'S FIRST CHARACTER
C
C IF ((KASE.LT.1).OR.(KASE.GT.26)) KASE=27      8 NOT ALPHA
C GO TO 1
C 0 401.402.403.404.405.406.407.408.409.410.
C 1 411.412.413.414.415.416.417.418.419.420.
C 2 421.422.423.424.425.426.427)
C 6 .KASE
C
C DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOMD TO BLANK
C
C 401 CONTINUE 8*** A
C 402 CONTINUE 8*** B
C GO TO 800
C
C 403 CONTINUE 8*** C
C IF (KOMD.EQ.'CHA') CALL DETCHA (KOMD)      8 CHANNEL (DETECTION)
C IF (KOMD.EQ.'CLE') CALL KMCLE (KOMD)      8 CLEAR
C IF (KOMD.EQ.'COL') CALL KMDCOL (KOMD)     8 COLOR
C IF (KOMD.EQ.'CRO') CALL KMDCRO (KOMD)    8 CROSSTAB
C GO TO 800
C
C 404 CONTINUE 8*** D
C IF (KOMD.EQ.'DEN') CALL KNODEN (KOMD)     8 DENSITY
    
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRD000
003

IF(KOND.EQ.'DIS') CALL PRDDIS(KOND) & DISPLAY
GO TO 800

C
405 CONTINUE 8000 E
IF(KOND.EQ.'EOF5') CALL PRDEXI(KOND) & END-OF-FILE CAUSES EXIT
IF(KOND.EQ.'EXI') CALL PRDEXI(KOND) & EXIT
IF(KOND.EQ.'EXP') CALL KMDEXP(KOND) & EXPLAIN
GO TO 800

C
406 CONTINUE 8000 F
IF(KOND.EQ.'FI ') CALL KMDFI (KOND) & FI (ENDIF)
GO TO 800

C
407 CONTINUE 8000 G
GO TO 800

C
408 CONTINUE 8000 H
IF(KOND.EQ.'HEA') CALL KMDHEA(KOND) & HEADING
GO TO 800

C
409 CONTINUE 8000 I
IF(KOND.EQ.'IF ') CALL KMDIF (KOND) & IF
IF(KOND.EQ.'INT') CALL KMDINT(KOND) & INTENSITY
GO TO 800

C
410 CONTINUE 8000 J
411 CONTINUE 8000 K
GO TO 800

C
412 CONTINUE 8000 L
IF(KOND.EQ.'LIS') CALL PRDLIS(KOND) & LIST
GO TO 800

C
413 CONTINUE 8000 M
IF(KOND.EQ.'MER') CALL KMDMER(KOND) & MERIDIAN
GO TO 800

C
414 CONTINUE 8000 N
IF(KOND.EQ.'NEW') CALL KMDNEW(KOND) & NEWS
IF(KOND.EQ.'NEX') CALL KMDNEX(KOND) & NEXT
GO TO 800

C
415 CONTINUE 8000 O
IF(KOND.EQ.'OFF') CALL KMDOFF(KOND) & OFF
IF(KOND.EQ.'ON ') CALL KMDON(KOND) & ON
IF(KOND.EQ.'ORI') CALL KMDORI(KOND) & ORIGIN
GO TO 800

C
416 CONTINUE 8000 P
IF(KOND.EQ.'PAO') CALL KMDPAO(KOND) & PAGE
IF(KOND.EQ.'PEE') CALL KMDPEE(KOND) & PEEK
IF(KOND.EQ.'PIC') CALL PRDPIC(KOND) & PICTURE
IF(KOND.EQ.'POK') CALL KMDPOK(KOND) & POKE
IF(KOND.EQ.'PRI') CALL KMDPRI(KOND) & PRINTER
GO TO 800

DAN PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

PRO000
 004

```

417 CONTINUE 0*** Q
    GO TO 000
C
418 CONTINUE 0*** R
    IF(KOMD.EQ.'RAD') CALL KMDRAD(KOMD)      0 RADIANCE
    IF(KOMD.EQ.'REN') CALL KMDREN(KOMD)     0 RENUMBER
    GO TO 000
C
419 CONTINUE 0*** S
    IF(KOMD.EQ.'SPA') CALL KMDSPA(KOMD)     0 SPACING
    IF(KOMD.EQ.'SYM') CALL KMDSYM(KOMD)    0 SYMBOLS
    GO TO 000
C
420 CONTINUE 0*** T
    IF(KOMD.EQ.'TAB') CALL KMDTAB(KOMD)     0 TABULATE
    IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD)    0 TIME
    GO TO 000
C
421 CONTINUE 0*** U
422 CONTINUE 0*** V
    GO TO 000
C
423 CONTINUE 0*** W
    IF(KOMD.EQ.'WIN') CALL KMDWIN(KOMD)    0 WINDOW
    GO TO 000
C
424 CONTINUE 0*** X
425 CONTINUE 0*** Y
    GO TO 000
C
426 CONTINUE 0*** Z
    IF(KOMD.EQ.'ZON') CALL KMDZON(KOMD)    0 ZONE
    GO TO 000
C
427 CONTINUE 0*** NOT ALPHABETIC
    IF(KOMD.EQ.'SAD') CALL KMDGAD(KOMD)    0 SADD
    IF(KOMD.EQ.'SAD') CALL KMDGAD(KOMD)    0 SADD
    IF(KOMD.EQ.'SAS') CALL KMDGAS(KOMD)    0 SASO
    IF(KOMD.EQ.'SBR') CALL KMDGBR(KOMD)    0 SBKPT
    IF(KOMD.EQ.'SFR') CALL KMDGFR(KOMD)    0 SFREE
    IF(KOMD.EQ.'SLO') CALL KMDGLO(KOMD)    0 SLOG
C
C
C IF COMMAND WAS NOT FOUND. TRY MACRO-COMMAND
C
000 IF(KOMD.NE.' ') KOMD='PRD-'           0 1ST 3 CHARS OF PROG NAME PLUS '--'
    IF(KOMD.NE.' ') CALL KMDXXX(KOMD)     0 MACRO COMMAND HANDLER
C
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
    IF(KOMD.NE.' ') CALL WARN$( 'INVALID COMMAND --')
C
C FORCE ALL FORTRAN I/O ROUTINES INTO SAME SEQ AS PRO000 (NEVER PERFORMED)
C
    IF(KOMD.EQ.'JUN.') READ(095,095) KOMD
  
```

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PR0000
005

895 FORMAT(IX)

C

C

C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY

C

RETURN
END

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRD129
001

```
      SUBROUTINE PRD129( 3 CALL PHASE 1/2/9 SUBROUTINES FOR PRDDET  
      I NANSUB)          3 NAME OF SUBROUTINE TO CALL  
      -----  
C  
C  
C (J C CRISP)  
C  
C  
C CALL PREVIOUSLY NAMED SUBROUTINE  
C  
      CALL TRACE  
      CALL NANSUB  
C  
C  
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY  
C  
      RETURN  
      END
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRD345
001

SUBROUTINE PRD345/ 8 CALL PHASE 3/4/5 SUBROUTINES FOR PRDDET
I NANSUB) 8 NAME OF SUBROUTINE TO CALL

C

C

C (J C CRISP)

C

C

C CALL PREVIOUSLY NAMED SUBROUTINE

C

CALL TRACE
CALL NANSUB

C

C

C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY

C

RETURN
END

SUBROUTINE PRODIS(3 DISPLAY DETECTION FILE(S) (PHASE 0)
U KOMO) 3 1: FIRST 3 CHARS OF COMMAND 0: SPACES

C

C

C

C HISTORY

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

E M SCHLOSSER LEC 10/05/75 ORIGINAL CODE
E M SCHLOSSER LEC 08/20/78 DELETE RETN K & MAKE EXTERNAL SUB
E M SCHLOSSER LEC 03/12/79 RADIANCE & CLASS DETECTION FILES
J C CRISP LEC 12/21/79 SPLIT OUT PRODIS & PRODI9

C METHOD

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

CALIBRATE SYMBOLS/SPACING/WINDOW. OPEN PRINT FILES. DISPLAY HEADINGS.
THEN NAME PRODIS TO GENERATE BODY OF DISPLAY.

C MACHINE-DEPENDENT CODE

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

NONE.

C EXTERNAL REFERENCES

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

GET5KH 3 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GET5IN 3 GET INTEGER DATA FIELD FROM UNIT 5
MDWARN 3 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
MDNOTE 3 PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
NVIATO 3 NAME 'VIA' 'TO' SUBROUTINES
CALSYM 3 CALIBRATE SYMBOL TABLE FOR OVERPRINTING
CALSPA 3 CALIBRATE TRANSFORMATION COEFFICIENTS FOR SPACING
CALWIN 3 CALIBRATE WINDOW ENVELOPES
OPRPRD 3 OPEN ALTERNATE PRINT FILE(S)
IDERT 3 IDENTIFY ERTS SCENE
IDCPRO 3 IDENTIFY CURRENT COMMAND SPECS FOR PRIDET
MDUNIT 3 WRITE HEADING LINE(S) AT TOP OF NEXT PAGE
WARN5 3 SUBMIT WARNING FOR MISSING/INVALID FIELD FROM UNIT 5
EXTERNAL PR000. NULSUB
EXTERNAL PR0129. PRODI9
EXTERNAL PR0345. PRODIS

C EXCEPTIONS

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

1. 'DISPLAY' MAY NOT BE A DEFAULT COMMAND.
2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE DISPLAY.
3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS

DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

PROD19
 002

```

C
C          CONDITION          ACTION          DIAGNOSTIC
C          -----          -
C
C PROCESSING DEFAULT COMMANDS
C (NWNDOW=0)                NONE                WARNING
C EXTRA SPECIFICATION      NONE                WARNING
C LIMIT CHANNEL VALUE RANGE IS NULL
C (LCVLOI>LCVHII)          NONE                WARNING
C DEMAND RUN & OVERPRINTED SYMBOLS NO OVERPRINTING IN PRODI3 NOTE
C DATA/CHECKOUT MODE      'TO' ROUTINE IS NULSUB NONE
C WARNING(S) OR FATAL ERROR(S) 'TO' ROUTINE IS PRODI9 NONE
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST      % COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C INCLUDE KOMALT.LIST     % COMMON/DEFINE FOR ALTERNATE PRINT FILES
C INCLUDE KOMNER.LIST     % COMMON ERTS SCENE PARAMETERS
C INCLUDE KOMKLS.LIST     % COMMON CLASSIFICATION SPECTRAL LIMITS
C INCLUDE KOMFIT.LIST     % COMMON ADJUSTMENT/REGISTRATION PARAMETERS
C INCLUDE KOMTBL.LIST     % COMMON TICK/FREQ/FACTOR TABLE
C INCLUDE KOMSYF.LIST     % COMMON SYMBOL TABLE
C INCLUDE WINDEF.LIST     % DEFINE STRUCTURE OF WINDOW PACKETS
C INCLUDE KOMOWW.LIST     % COMMON OUTPUT WINDOW PACKETS
C INCLUDE NULCST.LIST     % DEFINE NULL CHARACTER STRING
C INCLUDE PRDDEF.LIST     % DEFINE PRYDET PARAMETERS
C
C
C LOCAL DECLARATIONS
C -----
C
C INTEGER KHTEMP          % TEMPORARY
C INTEGER INTMP           % TEMPORARY
C INTEGER NPRLIN,NPRCOL  % NUMBER OF PRINT LINES/COLUMNS IN DISPLAY
C INTEGER NPRCIF         % NUMBER OF PRINT COLUMNS IN ONE FILE
C INTEGER LUNALT         % LOGICAL UNIT NUMBER TO SPOOL PRINTER OUTPUT
C
C
C PROCEDURE
C -----
C
C CALL TRACE
C
C
C GET DISPLAY TYPE
C
C KTBLTY=' NUL'          % MARK OLD FREQ TABLE AS DESTROYED
C IF(NWNDOW.EQ.0) CALL MDWARN('INVALID DEFAULT COMMAND')
C CALL GETSKH(KHTEMP,(3), NULCST) % ALLOW (BUT IGNORE) OPTIONAL TYPE
C CALL NVIATO(' PRD345.PRODI3) % NEXT CALL IS TO PRODI3
C
C
C DRAIN SPECS FOR CURRENT COMMAND
C

```

```

300 CALL GETSIN(INTERP,  +1,-1,'EXTRA DISPLAY SPECIFICATION --')
C
C
C CHECK RADIANCE LIMITS
C
      IF(LCVLO1.GT.LCVH1) CALL MDWARN(  'NO RADIANCE LIMITS')
      IF(MDATA.C.NE.0) GO TO 900      & DATA/CHECKOUT MODE
C
C
C CALIBRATE SYMBOLS/SPACING/WINDOW
C
      CALL CALSYM
      CALL CALSPA
      CALL CALWIN( 0.1)
C
C
C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
      IF(INDTOTL.NE.0) GO TO 900
      IF(NWINDOW.LT.0) CALL OPRPRD      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
      NWINDOW=IABS(NWINDOW)
      NPAGE=0
C
C
C COMPUTE SIZE OF PRINT WINDOW
C
      NPRLIN=IFIX(PPDOWH(WLIN.WMAX))-IFIX(PPDOWH(WLIN.WMIN))+1
      NPRCOL=IFIX(PPDOWH(WCOL.WMAX))-IFIX(PPDOWH(WCOL.WMIN))+1
      NPRCIF=KPAGE-6      & NUMBER OF PRINT COLUMNS PER FILE EXCLUDING
                          & LEFT MARGIN AND 2 NEAT LINE CHARS
C
C
C CHECK WIDTH OF PRINT WINDOW
C
      NITMAX=(1+(NPRCOL)/NPRCIF)
      IF(NITMAX.GT.MALTM) CALL MDWARN(  'WINDOW TOO WIDE')
C
C
C CHECK FOR DIAGNOSTICS
C
      IF(INDTOTL.NE.0) GO TO 900
      IF(IMBATCH.EQ.0).AND.(NCISYM.NE.1) CALL MDNOTE(
      * 'SYMBOLS NOT OVERPRINTED ON DEMAND TERMINAL')
C
C
C PRINT WINDOW HEADING FOR UNIT 6
C
      WRITE(6,415) NWINDOW,MTERAL
      415 FORMAT(' WINDOW NUMBER ',J3.0X,' DISPLAY ',8X.4A6)
      CALL IDERT( 6)
      CALL IDCPRD( 6)
C
C
C PRINT WINDOW HEADING FOR ALTERNATE PRINT FILE(S)
C
      LUNALT=10
  
```

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROD15
004

```
      DO 480 N=1,NPCOL,NPCIF
        CALL MDUNIT( 4,LUNALT)
        WRITE(LUNALT,415) NMNDOM,MTERAL
        CALL IDERT( LUNALT)
        CALL IDCPRD( LUNALT)
        LUNALT=LUNALT+1
480 CONTINUE
C
C
C ANY DIAGNOSTICS???
C
900 IF(NDTOTL.EQ.0) GO TO 990
      IF(MDATAC.NE.0) CALL NVIATOC( PRD000,NULSUB)      8 DATA/CHECKOUT
      IF(MDATAC.EQ.0) CALL NVIATOC( PRD129,PROD19)
C
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KOMD= ' '
      RETURN
      END
```


C	FROM	NEAT LINE CHAR	PRINT CELL SYMBOLS	DIAGNOSTIC	ACTION
C	READLN				
C	'EOF'	'.'	NO DATA (':')	NONE	PRINT LINE
C	'BADR'	'7'	NO DATA (':')	NONE	PRINT LINE
C	'BADF'	N/A	N/A	FATAL	RETURN
C	'OFL'	N/A	N/N	FATAL	RETURN

C GLOBAL DECLARATIONS

C -----

C INCLUDE KOMXQT.LIST § COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
 C INCLUDE KOMKLS.LIST § COMMON CLASSIFICATION INFO
 C INCLUDE KOMSYM.LIST § COMMON SYMBOL TABLE
 C INCLUDE WINDEF.LIST § DEFINE STRUCTURE OF WINDOW PACKETS
 C INCLUDE KOMOHV.LIST § COMMON OUTPUT WINDOW PACKETS
 C INCLUDE KOMTBL.LIST § COMMON TICK/FREQ TABLE
 C INCLUDE PRODEF.LIST § DEFINE PRDDET PARAMETERS
 C INCLUDE PXBDEF.LIST § DEFINE PIXEL BUFFER STRUCTURE
 C INCLUDE MAXINT.LIST § MAXIMUM INTEGER VALUE

C LOCAL DECLARATIONS

C -----

C PARAMETER NDBUFS=2 § NUMBER OF DETECTION BUFFERS
 C INTEGERS IN DET BUF = *INTS PREAMBLE + (*BINS+3)/4 + *EXTRA INTS
 C PARAMETER NWIDBF = 2*(PXBINS-1) + (3548+2+3)/4 + 27
 C WDS PRT BUF=WDS PREAM*(MAX FILES*(MAX COLS/PG-NEATLN COLS-MARG COLS))+4
 C PARAMETER NWIPBF=(PXBINS-1)*(MALTHI + (KPAQHI- 2 - 4)) + 4
 C
 C PARAMETER NFRQSZ=128 § * WDS PER CHANNEL IN LOCAL FREQ TABLE
 C PARAMETER NFRQCH=5 § * CHANNELS IN LOCAL FREQ TABLE
 C INTEGER NDTBUF(NWIDBF,NDBUFS) § ARRAY OF DETECTION BUFFERS
 C INTEGER NDBF § DETECTION BUFFER NUMBER
 C INTEGER NDBFHI § HIGHEST DETECTION BUFFER NUMBER TO USE
 C INTEGER (PRTBF(NWIPBF) § PRINT BUFFER
 C INTEGER NFREQ(NFRQSZ,NFRQCH) § LOCAL FREQUENCY TABLE (SCOPE INCLUDES
 C § INTERNAL ROUTINE RESSYM)
 C
 C INTEGER IPLIN § PRINT LINE
 C INTEGER IPCMIN,IPCMA § MINIMUM AND MAXIMUM PRINT COLUMN
 C INTEGER IPLMIN,IPLMA § MINIMUM AND MAXIMUM PRINT LINE
 C REAL ADJLIN,ADJSAM § ADJUSTED LINE AND SAMPLE
 C INTEGER ML100L,ML100H,ML100S § MSA LINE*100: LOW,HIGH,SPACING
 C INTEGER MSALIN § MSA LINE NUMBER
 C INTEGER MSASLO,MSASHI § LOW AND HIGH MSA SAMPLE
 C INTEGER ISTAT § I/O STATUS
 C INTEGER NTLCHR § NEAT LINE CHARACTER
 C INTEGER MROLCS,MRORCS § LEFT & RIGHT MARGIN CHAR STRING
 C INTEGER NPRLIN,NPRCOL § NUMBER OF PRINT LINES AND COLUMNS
 C INTEGER LASTLN § LAST LINE READ

C
 C
 C PROCEDURE


```

C -----
C
C     CALL TRACE
C
C
C     DETERMINE HOW MANY BUFFERS TO USE
C
C     NDBFMI=MIN0(NLIMCH,NDBUFS)
C     IF(NLIMCH.GT.NDBUFS) CALL MDNOTE(
C     & 'TOO MANY CHANNELS REQUESTED -- FIRST'.C084IN(NDBFMI,2),
C     & ' CHANNELS PROCESSED')
C
C
C     INITIALIZE MINIMUM AND MAXIMUM PRINT LINES AND COLUMNS
C
C     IPLMIN=PPDOHW(WLIN,WMIN)
C     IPLMAX=PPDOHW(WLIN,WMAX)
C     IPCHIN=PPDOHW(WCOL,WMIN)
C     IPCHAX=PPDOHW(WCOL,WMAX)
C
C
C     COMPUTE NUMBER OF PRINT LINES AND COLUMNS
C
C     NPRLIN=IPLMAX-IPLMIN+1
C     NPRCOL=IPCHAX-IPCHIN+1
C
C
C     CLEAR LOCAL FREQUENCY TABLE
C
C     DO 250 I=1,NLIMCH
C         DO 200 K=1,NFRQSZ
C             NFREQ(K,I)=0
C     200     CONTINUE
C     250     CONTINUE
C
C
C     INITIALIZE LOW AND HIGH LINES AND SPACING
C
C     CALL A4P (ADJLIN,ADJSAM,    FLOAT(IPLMIN),1.)
C     ML100L=ADJLIN*100.
C     CALL A4P (ADJLIN,ADJSAM,    FLOAT(IPLMAX),1.)
C     ML100H=ADJLIN*100.
C     ML100S=M5A0HW(WLIN,WSP100)
C
C
C     GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE BEFORE 1ST PRINT LINE
C
C     300 CALL SANSCL (IPRTBF,(IPLMIN-1),(IPCHIN),(IPCHAX))
C     IF ((NBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
C     &     CALL PROVFI (6,    '.4.' '.4.' '.0.' '.0.' '11111'.IPRTBF)
C     &     CALL PROVFI (10,   '.4.' '.4.' '.0.' '11111'.IPRTBF)
C
C
C     READ. MASK. RESAMPLE. SCREEN. SYMBOLIZE AND PRINT SCAN LINES
C
C     IPLIN=IPLMIN
  
```

```

LASTLN=-MAXINT
DO 350 ML100=ML100L,ML100H,ML100S
  MSALIN=ML100/100
  CALL A4P (ADJLIN,ADJSAN,   FLOAT(IPLIN),FLOAT(IPCHIN))
  MSASLO=ADJSAN
  CALL A4P (ADJLIN,ADJSAN,   FLOAT(IPLIN),FLOAT(IPCHAX))
  MSASHI=ADJSAN
  IF (MSALIN.EQ.LASTLN) GO TO 320      3 ALREADY IN BUFFER(S)
  NTLCHR=' '
  DO 310 NDBF=1,NDBFHI
    CALL READ2N(MDTBUF(1,NDBF),(NWIDBF),ISTAT,
               MSALIN,LINCH(NDBF),MSASLO,MSASHI)
    IF (ISTAT.EQ.'BADR') NTLCHR='?'
    IF ((ISTAT.NE.'BADF').AND.
        (ISTAT.NE.'OFL')) GO TO 310
    CALL M0FATL( CBS4CS(ISTAT,1,4),
               ' WHILE READING DETECTION FILE')
    CALL ERPRTA ('10          '.2,6,
               '*I/O ERROR - IGNORE OUTPUT*')
    GO TO 300
310      CONTINUE
320      LASTLN=MSALIN
    CALL MSKPIX (MDTBUF(1,1),   MDTBUF(1,1))
    IF ((MDTBUF(PXBINT,1).EQ.'BYT').OR.
        (MDTBUF(PXBINT,1).EQ.'CHR').OR.
        (MDTBUF(PXBINT,1).EQ.'INT').OR.
        (MDTBUF(PXBINT,1).EQ.'NUL')) GO TO 330
    CALL M0FATL('INVALID BIN TYPE ',
               CBS4CS(MDTBUF(PXBINT,1),1,3), ' IN PROD13')
    GO TO 300
330      IF (MDTBUF(PXBINT,1).EQ.'BYT') CALL RESSYM (IPRTBF,(IPLIN),
        (IPCHIN),(IPCHAX),   MDTBUF,(NWIDBF),(NDBUFS),GETBYT)
    IF (MDTBUF(PXBINT,1).EQ.'CHR') CALL RESSYM (IPRTBF,(IPLIN),
        (IPCHIN),(IPCHAX),   MDTBUF,(NWIDBF),(NDBUFS),GETICE)
    IF (MDTBUF(PXBINT,1).EQ.'INT') CALL RESSYM (IPRTBF,(IPLIN),
        (IPCHIN),(IPCHAX),   MDTBUF,(NWIDBF),(NDBUFS),GETINT)
    IF (MDTBUF(PXBINT,1).EQ.'NUL') CALL RESSYM (IPRTBF,(IPLIN),
        (IPCHIN),(IPCHAX),   MDTBUF,(NWIDBF),(NDBUFS),GETNUL)
    CALL CST4IN (MROLC,(1),(6),   MSALIN,4,'0')
    MRORC=MROLC
    IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
    CALL PROVFI (6,   MROLC,4,' '0,NTLCHR,'100000',IPRTBF)
    CALL PROVFI (10,   MROLC,4,MRORC,4,NTLCHR,'100000',IPRTBF)
    IPLIN=IPLIN+1
350 CONTINUE
C
C
C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE AFTER LAST PRINT LINE
C
    CALL SAMSCL (IPRTBF,(IPLMAX+1),(IPCHIN),(IPCHAX))
    IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
    CALL PROVFI (6,   '4.' '0.' '100000',IPRTBF)
    CALL PROVFI (10,   '4.' '4.' '111110',IPRTBF)
    CALL PROVFI (10,   '0000'4.'0000'4.' '311110',IPRTBF)
C
C

```

C MOVE DATA FROM LOCAL FREQ TABLE TO COMMON TABLE (REPLACING TICKS)

```
C
  KTBLTY='FREQ'
  KTBLNM='NHNDON'
  DO 600 I=1,NLIMCH
    DO 550 K=1,NFRQSZ
      KFREQ(K,1)=NFREQ(K,1)
  550   CONTINUE
  600   CONTINUE
  DO 700 I=1,10
    DO 650 K=1,15
      KFRCRO(I,K)=-MAXINT
  650   CONTINUE
  700   CONTINUE
```

```
C
C
C NEXT CALL IS TO PRDD19
```

```
C
  900 CALL NVIATO ( PRD129,PRDD19)
  RETURN
```

```
C
C
C
C
C
C
C
C
```

```
INTERNAL
SUBROUTINE SAMSCL (      8 GENERATE SAMPLE SCALE AND BORDER
  0 IPRTBF,      8 PRINT BUFFER
  1 IPLIN,      8 PRINT LINE
  1 IPCHIN,      8 MINIMUM PRINT COLUMN
  1 IPCMAX)      8 MAXIMUM PRINT COLUMN
```

```
C
C
C METHOD
```

```
C
  INITIALIZE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER
  PREAMBLE. ENCODE SAMPLE NUMBERS AND PUT COLON, STRING, AND
  COLON IN PRINT BUFFER.
```

```
C
C
C MACHINE-DEPENDENT CODE
```

```
C
  ASSUMES 8 CHARS PER INTEGER BIN
```

```
C
C
C EXTERNAL REFERENCES
```

```
C
  ANP      8 ADJUSTED COORD FOR PRINT/PLOT COORD
  PUTCHR   8 PUT CHAR IN CHAR STRING
  CST4IN   8 CHARACTER STRING FOR INTEGER
```

```
C
C
C GLOBAL DECLARATIONS
```

```
C
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROD13
000

```
C      INCLUDE KONOWM.LIST      & OUTPUT WINDOW PACKETS
C      INCLUDE WINDEF.LIST      & DEFINE WINDOW PACKETS
C      INCLUDE PXDEF.LIST      & DEFINE BUFFER STRUCTURE
C
C LOCAL DECLARATIONS
C
C      INTEGER IPRTBF(11)      & ARGUMENT
C      REAL ADJSAM      & ADJUSTED SAMPLE
C      INTEGER IPBIN      & POINTER TO PRINT BIN
C      INTEGER MSASAM      & SAMPLE NUMBER
C      INTEGER MS100L,MS100M,MS100S      & MSA SAMPLE*100: LOW,HIGH,SPACING
C
C C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
C      CALL A4P (ADJLIN,ADJSAM,   FLOAT(IPLIN),FLOAT(IPCHIN))
C      MS100L=ADJSAM*100.
C      CALL A4P (ADJLIN,ADJSAM,   FLOAT(IPLIN),FLOAT(IPCHAX))
C      MS100M=ADJSAM*100.
C      MS100S=MSAONH(MSAN,MSPI00)
C
C C SET PREAMBLE POINTERS AND BIN POINTER
C
C      IPRTBF(PXRECH)=0
C      IPRTBF(PXLINO)=IPLIN
C      IPRTBF(PXCHAN)=0
C      IPRTBF(PXQUAL)=0
C      IPRTBF(PXBINT)='INT'
C      IPRTBF(PXLBIN)=1
C      IPRTBF(PXLCOL)=IPCHIN
C      IPRTBF(PXHBIN)=IPCHAX-IPCHIN+1
C      IPRTBF(PXNCOL)=IPCHAX
C      IPRTBF(PXNOD1)=0
C      IPRTBF(PXNOD2)=0
C      IPRTBF(PXLJ01)=0
C      IPRTBF(PXHJ01)=0
C      IPBIN=IPRTBF(PXLBIN)-1
C
C C ENCODE SAMPLE NUMBER AND PUT COLON, STRING, AND COLON IN BUFFER
C
C      DO 100 MS100=MS100L,MS100M,MS100S
C          MSASAM=MS100/100
C          CALL PUTCHR (IPRTBF(PXBINS-IPBIN),11,   ':')
C          CALL CST4IN (IPRTBF(PXBINS-IPBIN),12,4,   MSASAM,4,'0')
C          CALL PUTCHR (IPRTBF(PXBINS-IPBIN),16,   ':')
C          IPBIN=IPBIN+1
C 100 CONTINUE
C
C
C      RETURN
```

```
C
C
C
C
C
C
C
C
C
C
INTERNAL
SUBROUTINE RESSYM (      ) RESAMPLE/SCREEN/COUNT FREQUENCY/SYMBOLIZE
O IPRBUF.      ) PRINT BUFFER
I IPLIN.      ) PRINT LINE
I IPCHIN.     ) MINIMUM PRINT COLUMN
I IPCHAX.     ) MAXIMUM PRINT COLUMN
.
I HDIBUF.     ) ARRAY OF DETECTION BUFFERS (ALL SAME BINTYPE)
I HWIDBF.     ) NUMBER OF WORDS IN ONE BUFFER
I NDBUFS.     ) NUMBER OF BUFFERS
I GETBINI    ) ROUTINE TO GET BIN VALUE--GETBYT.GETICE.GETINT.GETNUL
C
C
C METHOD
C
C COMPUTE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER PREAMBLE.
C FOR EACH SAMPLE, CHECK IF OUTSIDE OF INPUT WINDOW. CHECK IF OUT-
C SIDE OF RADIANCE LIMITS. COUNT FREQUENCY. AND SYMBOLIZE.
C
C
C EXTERNAL REFERENCES
C
C ANP          ) ADJUSTED MSS COORD FOR PRINT/PLOT COORD
C
C
C GLOBAL DECLARATIONS
C
C INCLUDE KOMOHV.LIST      ) COMMON OUTPUT WINDOW PACKETS
C INCLUDE KOMIBL.LIST     ) COMMON FREQ/TICK TABLE
C INCLUDE KOMKLS.LIST     ) COMMON CLASSIFICATION INFO
C INCLUDE PKBDEF.LIST     ) DEFINE BUFFER STRUCTURE
C INCLUDE KOMSYM.LIST     ) COMMON SYMBOL TABLE
C INCLUDE WINDEF.LIST     ) DEFINE WINDOW PACKETS
C
C
C LOCAL DECLARATIONS
C
C PARAMETER NUMBFS-2      ) NUMBER OF HPX BUFFERS
C INTEGER HDIBUF(HWIDBF,NDBUFS) ) ARGUMENT
C INTEGER IPRBUF(I)      ) ARGUMENT
C INTEGER NBINSD(NUMBFS) ) BIN NUMBER OF SAMPLE 0 FOR EACH HPX BUFFER
C INTEGER MS100L,MS100H,MS100S ) MSA SAMPLE*100: LOW,HIGH,SPACING
C INTEGER MNASAM          ) MSA SAMPLE NUMBER
C INTEGER IPB:N          ) PRINT BUFFER BIN POINTER
C REAL ADJSAM            ) ADJUSTED SAMPLE NUMBER
C INTEGER IPIXL1,IPIXL2  ) PIXEL VALUE FOR EACH BUFFER
C
C
C
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRD013
008

```
C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
  CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHIN))
  MS100L=ADJSAM*100.
  CALL A4P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCHMAX))
  MS100H=ADJSAM*100.
  MS100S=MSA0HW(HSAM,NSP100)
C
C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C
  DO 150 NUMBUF=1,NDBUFS
    NBINSD(NUMBUF)=MDTBUF(PXLBIN,NUMBUF)-MDTBUF(PXLSAM,NUMBUF)
  150 CONTINUE
C
C SET PREAMBLE POINTERS AND BIN POINTER
C
  IPRTBF(PXRECN)=MDTBUF(PXRECN,1)
  IPRTBF(PXLINO)=IPLIN
  IPRTBF(PXCHAN)=0
  IPRTBF(PXQUAL)=0
  IPRTBF(PXBINT)='INT'
  IPRTBF(PXLBIN)=2
  IPRTBF(PXLCOL)=IPCHIN
  IPRTBF(PXHBIN)=IPCHMAX-IPCHIN+2
  IPRTBF(PXHCOL)=IPCHMAX
  IPRTBF(PXNOIN)=0
  IPRTBF(PXNODA)=0
  IPRTBF(PXLJ01)=0
  IPRTBF(PXNJ01)=0
  IPBIN=IPRTBF(PXLBIN)-1
C
C RESAMPLE/SCREEN RADIANCE/COUNT FREQUENCY/LOOK UP SYMBOLS
C
  DO 400 MS100=MS100L,MS100H,MS100S
    MSASAM=MS100/100
C
C BUFFER 1
C
    IF ((MSASAM.LT.MDTBUF(PXLSAM,1)).OR.
      & (MSASAM.GT.MDTBUF(PXHSAM,1))) GO TO 350  & SAMPLE NOT IN BUFFER
    CALL GETBIN ((PIXL1,
      - MDTBUF(PXBINS,1),(MSASAM*NBINSD(1)))
    IF ((PIXL1.GE.MDTBUF(PXNODA,1)) GO TO 350  & NO DATA
      IF ((PIXL1.LT.LCVL01).OR.
      & ((PIXL1.GT.LCVH11)) GO TO 360  & OUT OF RAD LIMITS
      IF (NL1MCH-1.EQ.0) GO TO 320
C
C BUFFER 2
C
    IF ((MSASAM.LT.MDTBUF(PXLSAM,2)).OR.
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRO013
000

```
      (MSASAM.GT.MDTBUF(PXMSAM,2)) GO TO 350      & SAMPLE NOT IN BUFFER
      CALL GETBIN (PIXL2,
      MDTBUF(PXBINS,2),(MSASAM+NBINS0(2)))
      IF ((PIXL2.LT.LCVLO(2)).OR.
      (PIXL2.GT.LCVHI(2))) GO TO 360      & OUT OF RAD LIMITS
C
C COUNT FREQUENCY AND SYMBOLIZE
C
      NFREQ(PIXL2+1,2)=NFREQ(PIXL2+1,2)+1
320    NFREQ(PIXL1+1,1)=NFREQ(PIXL1+1,1)+1
      IPRTBF(PXBINS+IPBIN)=KSYM(PIXL1+1)
      GO TO 390
350    IPRTBF(PXBINS+IPBIN)=' '      & NO DATA SYMBOL
      GO TO 390
360    IPRTBF(PXBINS+IPBIN)=' '      & OUT OF RADIANCE LIMITS
390    IPBIN=IPBIN+1
400 CONTINUE
C
C
C
900 RETURN
      END
```


DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRO019
002

C ANY DIAGNOSTICS???

C

900 IF(NDFATL.EQ.0) GO TO 920

CALL HDNOTE('FATAL ERRORS -- NO DISPLAY GENERATED')
GO TO 990

920 IF(NDWARN.EQ.0) GO TO 960

CALL HDNOTE('PREVIOUS WARNINGS -- NO DISPLAY GENERATED')

IF(MBATCH.EQ.0) WRITE(6,925)

925 FORMAT(4X, '**TRY AGAIN!')

CALL HDCLR(NULCST)

GO TO 990

C

C

C PREPARE FOR NEXT DISPLAY

C

960 NWNDOW=NWNDOW+1

C

C

C RETURN TO NEXT STATEMENT IN CALLING ROUTINE

C

990 RETURN

END

SUBROUTINE PROEX1(3 TERMINATION ROUTINE FOR PRTOET
U KOND) 3 1: FIRST 3 CHARS OF COMMAND 0: SPACES

C
C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 08/12/75 DESIGN/CODE/TEST
C J C CRISP LEC 12/31/79 PRINT DISPLAYS ONSITE
C
C
C METHOD
C -----
C
C CONFIRM PROGRAM TERMINATION. ASK ABOUT PRINTER DISPLAYS. PRINT
C DISPLAYS ONSITE. IF REQUESTED. ELSE DELETE ALTERNATE PRINT FILES.
C TERMINATE PROGRAM.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C NONE
C
C
C EXTERNAL REFERENCES
C -----
C
C DLETPR 3 DELETE ALTERNATE PRINT FILES
C READ5 3 FILL BUFFER FOR UNIT 5
C CLOSPR 3 CLOSE (AND PRINT) ALTERNATE PRINT FILES
C PSTOP 3 PROGRAM TERMINATION
C
C
C EXCEPTIONS
C -----
C
C NONE
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST 3 COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
C INCLUDE KOMNER.LIST 3 COMMON ERTS SCENE PARAMETERS
C INCLUDE NULCST.LIST 3 DEFINE NULL CHARACTER STRING
C
C
C LOCAL DECLARATIONS
C -----
C
C INTEGER NORY 3 'N' OR 'Y' RESPONSE TO PRINT DISPLAYS ONSITE
C
C
C PROCEDURE

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROEX1
002

```
C -----  
C  
C      CALL TRACE  
C  
C  
C CONFIRM PROGRAM TERMINATION  
C  
C      WRITE(6,125)  
C      125 FORMAT(4X,'**PROGRAM TERMINATION**')  
C  
C  
C BATCH RUN WITH FATAL ERRORS OR ANY DEMAND RUN -- ASK ABOUT PRINTER DISPLAY(S)  
C  
C 150 IF((MBATCH.EQ.1).AND.(NOFATL.EQ.0)) GO TO 200      a BATCH & OK  
C      IF(NOFATL.NE.0) CALL MDWARN(  
C          'FATAL ERROR(S) ENCOUNTERED -- DISPLAY(S) ARE DEFECTIVE')  
C      WRITE(6,185)  
C 185  FORMAT(' OUTPUT DISPLAY(S) ON LINE PRINTER?')  
C      IF(KOMD.NE.'EOF5') CALL READ5(LSSTAT, ' ')  
C      NORY='N'  
C      CALL GET5KH(NORY,(1), NULCST)  
C      IF(NORY.NE.'Y') GO TO 700  
C  
C  
C PRINT DISPLAY(S) ONSITE  
C  
C 200 NWNDOW=MAX0(0,NWNDOW-1)  
C      WRITE(6,245) NWNDOW  
C 245  FORMAT(1X,14,' DISPLAY(S) PRINTED')  
C      CALL CLOSPR  
C      GO TO 900  
C  
C  
C DON'T PRINT DISPLAY(S) ONSITE  
C  
C 700 CALL DLETPR  
C  
C  
C TERMINATE PROGRAM  
C  
C 900 CALL PSTOP( NULCST)  
C  
C  
C PSTOP DOES NOT RETURN  
C  
C      END
```


DAH PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

PRDL15
 002

```

C
C PROCESSING DEFAULT COMMANDS
C (NHNDOW=0) NONE WARNING
C KLSTYP SPECIFICATION MISSING USE COMMON KLSTYP NONE
C EXTRA SPECIFICATION NONE WARNING
C LIMIT CHANNEL VALUE RANGE IS NULL
C (LCVLOI>LCVHI) NONE WARNING
C DATA/CHECKOUT MODE 'TO' ROUTINE IS NULSUB NONE
C WARNING(S) OR FATAL ERROR(S) 'TO' ROUTINE IS PRDL15 NONE
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST @ COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
C INCLUDE KOMALT.LIST @ COMMON DEFINE FOR ALTERNATE PRINT FILES
C INCLUDE KOMNER.LIST @ COMMON ERTS SCENE PARAMETERS
C INCLUDE KOMKLS.LIST @ COMMON CLASSIFICATION SPECTRAL LIMITS
C INCLUDE KOMTBL.LIST @ COMMON TICK/FREQ/FACTOR TABLE
C INCLUDE WINDEF.LIST @ DEFINE STRUCTURE OF WINDOW PACKETS
C INCLUDE KOMOWH.LIST @ COMMON OUTPUT WINDOW PACKETS
C INCLUDE NULCST.LIST @ DEFINE NULL CHARACTER STRING
C
C LOCAL DECLARATIONS
C -----
C
C INTEGER KHTEMP @ TEMPORARY
C INTEGER INTEMP @ TEMPORARY
C INTEGER NPRLIN @ NUMBER OF PRINT LINES IN LIST
C INTEGER NPRCOL @ NUMBER OF LOGICAL PPD COLUMNS IN LIST
C INTEGER LUNALT @ LOGICAL UNIT NUMBER TO SPOOL PRINTER OUTPUT
C INTEGER NPRCIF @ NUMBER OF LOGICAL PPD COLUMNS IN ONE FILE
C INTEGER NODMAX @ MAXIMUM # OF NODES IN PPDOWH
C INTEGER NOD @ # OF NODE IN PPDOWH/PPDOW
C INTEGER PPDOWH(2.HNO) @ TEMPORARY PPDOWH
C
C PROCEDURE
C -----
C
C CALL TRACE
C
C GET LIST TYPE
C
C KTBLTY=' NUL' @ MARK OLD FREQ TABLE AS DESTROYED
C IF(NHNDOW.EQ.0) CALL MDWARN('INVALID DEFAULT COMMAND')
C CALL GETSKH(KHTEMP,(3), NULCST)
C CALL NVIATO( PRD345,PRDL13) @ NEXT CALL IS TO PRDL13
C
C DRAIN SPECS FOR CURRENT COMMAND
C
C 300 CALL GETSIN(INTEMP, +1,-1,'EXTRA LIST SPECIFICATION ---')
C
  
```

```
C
C CHECK RADIANCE LIMITS
C
  IF(LCVLOI.GT.LCVHI) CALL MDWARN( 'NO RADIANCE LIMITS')
  IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE

C
C CALIBRATE SPACING
C
  CALL CALSPA

C
C SAVE THE PHYSICAL PPD VERTEX COLUMNS
C
  NODMAX=800L(PPDOWN(WUSED.WHEAD))
  DO 350 NOD=WVER.NODMAX
    PPDTON(WCOL.NOD)=PPDOWN(WCOL.NOD)
  350 CONTINUE

C
C TRANSFORM TO LOGICAL PPD VERTEX COLUMNS FROM PHYSICAL
C
  DO 375 NOD=WVER.NODMAX
    PPDOWH(WCOL.NOD)=PPDTON(WCOL.NOD)/3
  375 CONTINUE

C
C CALIBRATE OUTPUT WINDOW ENVELOPE USING LOGICAL PPD VERTEX COLUMNS
C
  CALL CALWIN( 0.)

C
C RESTORE THE PHYSICAL PPD VERTEX COLUMNS
C
  DO 400 NOD=WVER.NODMAX
    PPDOWH(WCOL.NOD)=PPDTON(WCOL.NOD)
  400 CONTINUE

C
C CHECK FOR DIAGNOSTICS
C
  IF (NDTOTL.NE.0) GO TO 900

C
C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
  IF(NHNDOW.LT.0) CALL OPRPRD      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
  NHNDOW=ABS(NHNDOW)
  NPAGE=0

C
C COMPUTE LOGICAL SIZE OF PRINT WINDOW FROM LOGICAL PPD ENVELOPE
C
  NPRLIN=IFIX(PPDOWN(WLIN.WMAX))-IFIX(PPDOWN(WLIN.WMIN))+1
  NPRCOL=IFIX(PPDOWN(WCOL.WMAX))-IFIX(PPDOWN(WCOL.WMIN))+1
  NPRCIF=(KPAGE-4-2)/3      & NUMBER OF LOGICAL PPD COLUMNS PER FILE
```

```
C                                     B EXCLUDING LEFT MARGIN & 2 NEAT LINES
C
C
C CHECK WIDTH OF PRINT WINDOW
C
      NITHAX=(1+(NPRCOL)/(NPRCIF))
      IF(NITHAX.GT.NALTM) CALL MDHARN( 'WINDOW TOO WIDE')
C
C
C CHECK FOR DIAGNOSTICS
C
      IF(NDTOTL.NE.0) GO TO 900
C
C
C PRINT WINDOW HEADING FOR UNIT B
C
      WRITE(6,415) NWNDOW.MTERAL
415 FORMAT(' WINDOW NUMBER ',J3.6X,'LIST',6X,4A6)
      CALL IDERT( 6)
      CALL IDCPRD( 6)
C
C
C PRINT WINDOW HEADING FOR EACH ALTERNATE PRINT FILE IN LOGICAL PPD ENVELOPE
C
      LUNALT=10
      DO 460 N=1,NPRCOL,NPRCIF
          CALL MDUNIT( 4,LUNALT)
          WRITE(LUNALT,415) NWNDOW.MTERAL
          CALL IDERT( LUNALT)
          CALL IDCPRD( LUNALT)
          LUNALT=LUNALT+1
460 CONTINUE
C
C
C ANY DIAGNOSTICS???
C
900 IF(NDTOTL.EQ.0) GO TO 990
      IF(MDATAC.NE.0) CALL NVIATOC( PRD000.NULSUB) B DATA/CHECKOUT
      IF(MDATAC.EQ.0) CALL NVIATOC( PRD129.PRDL19)
C
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KOMD='
      RETURN
      END
```

SUBROUTINE PROL13 8 LIST DETECTION FILE(S) (PHASE 3)

C
C

C HISTORY
C -----

C	J C CRISP	LEC	09/21/79	REQUIREMENTS
C	J C CRISP	LEC	10/13/79	ALGORITHM DESIGN
C	J C CRISP	LEC	12/26/79	ALGORITHM CODING
C	J C CRISP	LEMSCO	05/16/80	USE CHANNEL(S) SPECIFIED BY LIMCH
C	J C CRISP	LEMSCO	08/10/80	ADD 4 WORDS TO PRINT BUFFER

C METHOD
C -----

C INITIALIZE LOW AND HIGH PRINT LINES AND COLUMNS. GENERATE
C AND PRINT TOP SAMPLE SCALE AND BORDER. INITIALIZE LOW AND
C HIGH ADJUSTED LINE. CALL READEN TO READ LINE. MASK NON-
C TRIVIAL WINDOW. RESAMPLE/SCREEN/COUNT FREQUENCY.
C GENERATE AND PRINT BOTTOM SAMPLE SCALE AND BORDER.
C NAME PROL13 AS 'TO' ROUTINE FOR WRAP-UP OF LIST
C PROCESSING.

C MACHINE-DEPENDENT CODE
C -----

C ASSUMES 8 CHARS TO AN INTEGER BIN.

C EXTERNAL REFERENCES
C -----

C	AWP	8 ADJUSTED COORD FOR PRINT/PLOT COORD
C	READ2N	8 READ SCAN LINE FROM DETECTION FILE(S)
C	GETCHR	8 GET CHARACTER FROM CHARACTER STRING
C	MSKPIX	8 MASK NON-TRIVIAL WINDOW
C	PROVFI	8 PRINT/OVERPRINT FILES
C	MDFATL	8 PRINT/LOG/COUNT 'FATAL ERROR' MESSAGES
C	NVIATO	8 NAME 'VIA' 'TO' SUBROUTINES
C	ERPRTA	8 WRITE TO ALTERNATE PRINT FILES
C	CST4IN	8 CHARACTER STRING FOR INTEGER
C	DOUBLE PRECISION CBS4CS	8 VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH
C	DOUBLE PRECISION CBS4IN	8 VARIABLE-LENGTH CHAR STRING FOR INTEGER
C	EXTERNAL PRD129.	PROL13
C	EXTERNAL RESCRN.	GETBYT.GETICE.GETINT.GETNUL 8 ROUTINE TO GET BIN VALUE

C EXCEPTIONS
C -----

C	STATUS				
C	FROM				
C	READ2N	NEAT LINE CHAR	PRINT CELL SYMBOLS	DIAGNOSTIC	ACTION

DAM PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

PROL13
 002

C	'EOF'	'..'	NO DATA (':::')	NONE	PRINT LINE
C	'BADR'	'?'	NO DATA (':::')	NONE	PRINT LINE
C	'BADF'	N/A	N/A	FATAL	RETURN
C	'OFL'	N/A	N/A	FATAL	RETURN

C GLOBAL DECLARATIONS
 C -----
 C

```

INCLUDE KOMXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST      & COMMON CLASSIFICATION INFO
INCLUDE WINDEF.LIST      & DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMONW.LIST      & COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMTBL.LIST      & COMMON TICK/FREQ TABLE
INCLUDE PXDEF.LIST       & DEFINE BUFFER STRUCTURE
INCLUDE PRODEF.LIST      & DEFINE PRIDET PARAMETERS
INCLUDE MAXINT.LIST      & MAXIMUM INTEGER VALUE
  
```

C
 C
 C LOCAL DECLARATIONS
 C -----
 C

```

PARAMETER NOBUFS=2      & # OF DETECTION BUFFERS IN ARRAY

INTEGERS IN DET BUF = #INTS PREAMBLE + (#BINS-3)/4 + #EXTRA INTS
PARAMETER NWIDBF = 2*(PXBINS-1) + (3540-2-3)/4 + 27

WDS PRT BUF=WDS PREAM+(MAX FILES*(MAX COLS/PO-NEATLN COLS-MARG COLS))*4
PARAMETER NWIPBF=(PXBINS-1)*(MALTHI + (KPAQMI - 2 - 4)) * 4

PARAMETER NFRQSZ=128    & # WDS PER CHANNEL IN LOCAL FREQ TABLE
PARAMETER NFRQCH=9      & # CHANNELS IN LOCAL FREQ TABLE
INTEGER NDTBUF(NWIDBF,NOBUFS) & ARRAY OF DETECTION BUFFERS
INTEGER NDBF             & DETECTION BUFFER NUMBER
INTEGER NDBFHI           & HIGHEST DETECTION BUFFER NUMBER TO USE
INTEGER IPRTBF(NWIPBF)  & PRINT BUFFER
INTEGER NFRQ(NFRQSZ,NFRQCH) & LOCAL FREQUENCY TABLE (SCOPE INCLUDES
                                & INTERNAL ROUTINE RE'SCRNI)

INTEGER IPLIN           & PRINT LINE
INTEGER IPCHIN,IPCHAX   & MINIMUM, MAXIMUM PRINT COLUMN
REAL ADJLN,ADJSAM       & ADJUSTED LINE AND SAMPLE
INTEGER ML100L,ML100M,ML100S & MSA LINE*100: LOW,HIGH,SPACING
INTEGER MSALIN          & MSA LINE NUMBER
INTEGER ISTAT           & I/O STATUS
INTEGER NTLCHR          & NEAT LINE CHARACTER
INTEGER MROLC,MROPCS    & LEFT & RIGHT MARGIN CHAR STRING
INTEGER IPLMIN,IPLMAX   & MINIMUM AND MAXIMUM PRINT LINE
INTEGER ML100           & MSA LINE*100
INTEGER MSASLO,MSASHI   & LOW & HIGH MSA SAMPLE NUMBER
INTEGER NPRLIN,NPRCOL   & # PRINT LINES, COLUMNS
INTEGER LASTLN         & LAST LINE READ
  
```

C
 C
 C PROCEDURE
 C -----


```

DO 350 NL100=NL100L,NL100H,NL100S
  NSALIN=NL100/100
  CALL A4P (ADJLIN,ADJSAM,   FLOAT((IPLIN),FLOAT(IPCHIN))
  NSASLO=ADJSAM
  CALL A4P (ADJLIN,ADJSAM,   FLOAT((IPLIN),FLOAT(IPCHAX))
  NSASHI=ADJSAM
  IF(NSALIN.EQ.LAS(L,N)) GO TO 320      8 ALREADY IN BUFFER(S)
  NTLCHR=' '
  DO 310 NDBF=1,NDBFH1
    CALL READN(MDTBUF(1,NDBF),(NWIDTH),ISTAT,
      NSALIN,LINCH(NDBF),NSASLO,NSASHI)
    IF(ISTAT.EQ.'BADR') NTLCHR='?'
    IF((ISTAT.NE.'BADF').AND.
      (ISTAT.NE.'OFL')) GO TO 310
    CALL HDFATL(   COS4CS(ISTAT,1,4),
      ' WHILE READING DETECTION FILE')
    CALL ERPRTA ('10          '.2,8,
      '*I/O ERROR - IGNORE OUTPUT*')
    GO TO 300
  310      CONTINUE
  LASTLN=NSALIN
  320      CALL HSKPIX (MDTBUF(1,1),   MDTBUF(1,1))
  IF ((MDTBUF(PXBINT,1).EQ.'BYT').OR.
    (MDTBUF(PXBINT,1).EQ.'CHR').OR.
    (MDTBUF(PXBINT,1).EQ.'INT').OR.
    (MDTBUF(PXBINT,1).EQ.'NUL')) GO TO 330
  CALL HDFATL('INVALID BIN TYPE ',
    COS4CS(MDTBUF(PXBINT,1),1,3),' IN PROL13')
  GO TO 300
  330      IF (MDTBUF(PXBINT,1).EQ.'BYT') CALL RESCRN (IPRTBF,(IPLIN),
    (IPCHIN),(IPCHAX),   MDTBUF,(NWIDTH),(NDBUFS).GETBYT)
  IF (MDTBUF(PXBINT,1).EQ.'CHR') CALL RESCRN (IPRTBF,(IPLIN),
    (IPCHIN),(IPCHAX),   MDTBUF,(NWIDTH),(NDBUFS).GETICE)
  IF (MDTBUF(PXBINT,1).EQ.'INT') CALL RESCRN (IPRTBF,(IPLIN),
    (IPCHIN),(IPCHAX),   MDTBUF,(NWIDTH),(NDBUFS).GETINT)
  IF (MDTBUF(PXBINT,1).EQ.'NUL') CALL RESCRN (IPRTBF,(IPLIN),
    (IPCHIN),(IPCHAX),   MDTBUF,(NWIDTH),(NDBUFS).GETNUL)
  CALL CST4IN (MROLCS,(1),(8),   NSALIN,4,'0')
  MRRORCS=MROLCS
  IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
    CALL PROVFI (8,MROLCS,4,' '0,NTLCHR,'1*****',IPRTBF)
  CALL PROVFI (10,MROLCS,4,MRRORCS,4,NTLCHR,'1*****',IPRTBF)
  IPLIN=IPLIN+1
  350 CONTINUE
  C
  C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE AFTER LAST PRINT LINE
  C
  CALL SANSCL (IPRTBF,(IPLMAX+1),(IPCHIN),(IPCHAX))
  IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
    CALL PROVFI (8,' '4,' '0,' '1*****',IPRTBF)
  CALL PROVFI (10,' '4,' '4,' '11111',IPRTBF)
  CALL PROVFI (10,'*****4*****4,' '31111',IPRTBF)
  C
  C MOVE DATA FROM LOCAL FREQ TABLE TO COMMON TABLE

```


DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROL13
006

```
C
C   A*P       3 ADJUSTED COORD FOR PRINT/PLOT COORD
C   PUTCHR    3 PUT CHAR IN CHAR STRING
C   CST4IN    3 CHARACTER STRING FOR INTEGER
C
C EXCEPTIONS
C
C   ASSUMES 6 CHARACTERS PER INTEGER BIN.
C
C GLOBAL DECLARATIONS
C
C   INCLUDE KOMOHV.LIST      3 COMMON OUTPUT WINDOW PACKETS
C   INCLUDE WINDEF.LIST     3 DEFINE STRUCTURE OF WINDOW PACKETS
C   INCLUDE PXBDEF.LIST     3 DEFINE PIXEL BUFFER STRUCTURE
C
C LOCAL DECLARATIONS
C
C   INTEGER IPRTBF(1)       3 ARGUMENT
C   REAL ADJLIN,ADJSAM      3 ADJUSTED LINE, SAMPLE
C   INTEGER IPBIN           3 POINTER TO PRINT BIN
C   INTEGER MSASAM         3 SAMPLE NUMBER
C   INTEGER MS100L,MS100H,MS100S  3 MSA SAMPLE*100: LOW,HIGH,SPACING
C
C
C C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
C   CALL A*P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCMIN))
C   MS100L=ADJSAM*100.
C   CALL A*P (ADJLIN,ADJSAM,  FLOAT(IPLIN),FLOAT(IPCMAX))
C   MS100H=ADJSAM*100.
C   MS100S=MSAONH(MSAM,WSP100)
C
C
C C SET PREAMBLE POINTERS
C
C   IPRTBF(PXRECN)=0
C   IPRTBF(PXLINO)=IPLIN
C   IPRTBF(PXCHAN)=0
C   IPRTBF(PXQUAL)=0
C   IPRTBF(PXBINT)='INT'
C   IPRTBF(PXLBIN)=2
C   IPRTBF(PXLCOL)=IPCMIN
C
C CONVERT LOGICAL PPD COLUMNS TO PHYSICAL PPD BINS
C   IPRTBF(PXHBIN)=(IPCMAX-IPCMIN+1)*3+1
C
C
C   IPRTBF(PXHCOL)=IPCMAX
C   IPRTBF(PXNOIN)=0
C   IPRTBF(PXNODA)=0
C   IPRTBF(PXLJOI)=0
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROL13
000

```
C
C MACHINE-DEPENDENT CODE
C
C     NONE.
C
C EXTERNAL REFERENCES
C
C     A4P           3 ADJUSTED MSS COORD FOR PRINT/PLOT COORD
C     CST4IN       3 CHARACTER STRING FOR INTEGER
C     GETICE       3 GET CHARACTER FROM CHARACTER STRING
C
C
C EXCEPTIONS
C
C     NONE.
C
C GLOBAL DECLARATIONS
C
C     INCLUDE KOMOHV.LIST      3 COMMON OUTPUT WINDOW PACKETS
C     INCLUDE KOMTBL.LIST     3 COMMON FREQ/TICK TABLE
C     INCLUDE KOMKLS.LIST     3 COMMON CLASSIFICATION INFO
C     INCLUDE PXBDEF.LIST     3 DEFINE BUFFER STRUCTURE
C     INCLUDE KOMSYH.LIST     3 COMMON SYMBOL TABLE
C     INCLUDE WINDEF.LIST     3 DEFINE STRUCTURE OF WINDOW PACKETS
C
C LOCAL DECLARATIONS
C
C     PARAMETER NUMBFS=2
C     INTEGER MOTBUF(NHIDBF,NDBUFS)      3 ARGUMENT
C     INTEGER IPRTF(I)                  3 ARGUMENT
C     INTEGER IPIXL1,IPIXL2              3 VALUE OF PIXEL FOR EACH MPX BUFFER
C     INTEGER NBINSO(NUMBFS)             3 BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER
C     INTEGER MS100L,MS100H,MS100S       3 MSA SAMPLE*100: LOW,HIGH,SPACING
C     INTEGER MSASAM                      3 MSA SAMPLE NUMBER
C     INTEGER IPBIN                        3 PRINT BUFFER BIN POINTER
C     REAL ADJLIN,ADJSAM                   3 ADJUSTED LINE AND SAMPLE
C     INTEGER MS100                        3 MSA SAMPLE*100
C     INTEGER KSTPIX                       3 PIXEL VALUE CHARACTER STRING
C
C
C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
C     CALL A4P (ADJLIN,ADJSAM,   FLOAT(IPLIN),FLOAT(IPCHIN))
C     MS100L=ADJSAM*100.
C     CALL A4P (ADJLIN,ADJSAM,   FLOAT(IPLIN),FLOAT(IPCHAX))
C     MS100H=ADJSAM*100.
C     MS100S=MSA0WH/(WSAM.WSP100)
C
C
C
```

```
C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C
  DO 150 NUMBUF=1,NBUBFS
    NBINS0(NUMBUF)=MDTBUF(PXLBIN,NUMBUF)-MDTBUF(PXLSAM,NUMBUF)
  150 CONTINUE
C
C
C SET PREAMBLE POINTERS AND BIN POINTER
C
  IPRTBF(PXRECN)=MDTBUF(PXRECN,1)
  IPRTBF(PXLINO)=1PLIN
  IPRTBF(PXCHAN)=0
  IPRTBF(PXQUAL)=0
  IPRTBF(PXBINT)='INT'
  IPRTBF(PXLBIN)=2
  IPRTBF(PXLCOL)=1PCMIN
C
C CONVERT LOGICAL PPD COLUMNS TO PHYSICAL PPD BINS
  IPRTBF(PXHBIN)=((1PCMAX-1PCMIN+1)*3)+1
C
  IPRTBF(PXHCOL)=1PCMAX
  IPRTBF(PXNOIN)=0
  IPRTBF(PXNODA)=0
  IPRTBF(PXLJO1)=0
  IPRTBF(PXHJO1)=0
  IPBIN=IPRTBF(PXLBIN)-1
C
C
C RESAMPLE/SCREEN RADIANCE/COUNT FREQUENCY
C
  DO 400 MS100=MS100L,MS100H,MS100S
    MSASAM=MS100/100
C
C RESAMPLE AND SCREEN BUFFER 1
C
  IF ((MSASAM.LT.MDTBUF(PXLSAM,1)).OR.
    & (MSASAM.GT.MDTBUF(PXHSAM,1))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPXL1,
    & MDTBUF(PXBINS,1),(MSASAM+NBINS0(1)))
  IF ((IPXL1.GE.MDTBUF(PXNODA,1)) GO TO 350  & NO DATA
    IF ((IPXL1.LT.LCVLJ1).OR.
    & (IPXL1.GT.LCVH1)) GO TO 360  & OUT OF RAD LIMITS
    IF (NLIMCH-1.EQ.0) GO TO 320
C
C BUFFER 2
C
  IF ((MSASAM.LT.MDTBUF(PXLSAM,2)).OR.
    & (MSASAM.GT.MDTBUF(PXHSAM,2))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPXL2,
    & MDTBUF(PXBINS,2),(MSASAM+NBINS0(2)))
  IF ((IPXL2.LT.LCVL0(2)).OR.
    & (IPXL2.GT.LCVH1(2))) GO TO 360  & OUT OF RAD LIMITS
C
C COUNT FREQUENCY
C
  NFREQ(IPXL2+1,2)=NFREQ(IPXL2+1,2)+1
```



```
320      NFREQ(IPXL1+1,1)-NFREQ(IPXL1+1,1)+1
C
C
C INSERT RADIANCE VALUE INTO PRINT BUFFER
C      (3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN)
C
      CALL CST4IN(KSTPIX,(1),3,
      *      IPXL1,3,'0')
      CALL GETCHR(IPRTBF(PXBINS+IPBIN+0),      * HUNDREDS DIGIT
      *      KSTPIX,(1))
      CALL GETCHR(IPRTBF(PXBINS+IPBIN+1),      * TENS DIGIT
      *      KSTPIX,(2))
      CALL GETCHR(IPRTBF(PXBINS+IPBIN+2),      * UNITS DIGIT
      *      KSTPIX,(3))
      IF(IPRTBF(PXBINS+IPBIN+0).EQ.'0') IPRTBF(PXBINS+IPBIN+0)='.'
      GO TO 390
C
C
C INSERT 'NO DATA' SYMBOLS INTO PRINT BUFFER
C
350      IPRTBF(PXBINS+IPBIN+0)='.'
      IPRTBF(PXBINS+IPBIN+1)='.'
      IPRTBF(PXBINS+IPBIN+2)='.'
      GO TO 390
C
C
C INSERT 'NO INFO' SYMBOLS INTO PRINT BUFFER
C
360      IPRTBF(PXBINS+IPBIN+0)='.'
      IPRTBF(PXBINS+IPBIN+1)='.'
      IPRTBF(PXBINS+IPBIN+2)='.'
390      IPBIN=IPBIN+3
400 CONTINUE
C
C
C RETURN TO CALLING ROUTINE
C
900 RETURN
      END
```


DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRDL19
002

```
C
C ANY DIAGNOSTICS???
C
900 IF(INDFATL.EQ.0) GO TO 920
    CALL MDNOTE( 'FATAL ERRORS -- NO LIST GENERATED')
    GO TO 990
920 IF(INDWARN.EQ.0) GO TO 980
    CALL MDNOTE( 'PREVIOUS WARNINGS -- NO LIST GENERATED')
    IF(MBATCH.EQ.0) WRITE(6,925)
925  FORMAT(4X,'**TRY AGAIN!')
    CALL MDCLR( NULCST)
    GO TO 990
C
C
C PREPARE FOR NEXT WINDOW
C
980 NNND00H=NNND0W+1
C
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
990 RETURN
    END
```


C 2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE PICTURE.
 C
 C 3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS

CONDITION	ACTION	DIAGNOSTIC
PROCESSING DEFAULT COMMANDS		
(INWINDOW=0)	NONE	WARNING
KLSTYP IN COMMON UNDEFINED	KLSTYP:='RAD'	NONE
KLSTYP SPECIFICATION MISSING	USE COMMON KLSTYP	NONE
KLSTYP SPECIFICATION INVALID	NONE	WARNING
EXTRA SPECIFICATION	NONE	WARNING
LIMIT CHANNEL VALUE RANGE IS NULL		
(LCVLO>LCVHI)	NONE	WARNING
COLOR MODE SWITCH NOT ON	NONE	WARNING
BATCH RUN	NONE	WARNING
DATA/CHECKOUT MODE	'TO' ROUTINE IS NULSUB	NONE
WARNING(S) OR FATAL ERROR(S)	'TO' ROUTINE IS PROPI9	NONE

C GLOBAL DECLARATIONS

INCLUDE KOMXQT.LIST	% COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST	% COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST	% COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMTBL.LIST	% COMMON TICK/FREQ/FACTOR TABLE
INCLUDE KOMSYH.LIST	% COMMON SYMBOL TABLE
INCLUDE KOMKS.LIST	% COMMON COLOR SCREEN PARAMETERS
INCLUDE WINDEF.LIST	% DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMOHV.LIST	% COMMON OUTPUT WINDOW PACKETS
INCLUDE NULCST.LIST	% DEFINE NULL CHARACTER STRING

C LOCAL DECLARATIONS

INTEGER KHTMP	% TEMPORARY
INTEGER INTEMP	% TEMPORARY
INTEGER NPRLIN,NPRCOL	% NUMBER OF PPD LINES/COLUMNS IN PICTURE

C PROCEDURE

C -----
 C CALL TRACE
 C
 C CHECK IF COMMAND IS LEGAL

IF(INWINDOW.EQ.0) CALL MDWARN('INVALID DEFAULT COMMAND')
 IF(BATCH.NE.0) CALL MDWARN('PICTURE COMMAND NOT ALLOWED IN BATCH')
 IF(COLOR.EQ.0) CALL MDWARN('PICTURE COMMAND NOT ALLOWED (COLOR MODE NOT ON)')

```
C
C
C GET PICTURE TYPE
C
  KTBLTY=' NUL '      & MARK OLD FREQ TABLE AS DESTROYED
  CALL GETSKH(KHTEMP,(3), NULCST) & ALLOW (BUT IGNORE) OPTIONAL PICT TYPE
  CALL NVIATO( PRD345,PROPI3) & NEXT CALL IS TO PROPI3
C
C DRAIN SPECS FOR CURRENT COMMAND
C
  300 CALL GETSIN(INTERP.  +1.-1.'EXTRA PICTURE SPECIFICATION --')
C
C CHECK RADIANCE LIMITS
C
  IF(LCVLO1.OT.LCVH11) CALL MDWARN( 'NO RADIANCE LIMITS')
  IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE
C
C CALIBRATE COLORS/SPACING/WINDOW
C
  CALL CALCOL
  CALL CALSPA
  CALL CALWIN( 0.)
C
C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
  IF(INDTOTL.NE.0) GO TO 900
  IF(NHNDOW.LT.0) CALL OPRPRD      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
  NHNDOW=1ABS(NHNDOW)
  NPAGE=0
C
C CROP OUTPUT WINDOW TO FIT INPUT WINDOW & COLOR SCREEN
C
  CALL CROPOW( KSLINE,KSCOLM)
  IF(INDTOTL.NE.0) GO TO 900
C
C COMPUTE SIZE OF PRINT/PLOT DEVICE (COLOR SCREEN) WINDOW
C
  NPRLIN=IFIX(PPDOHW(WLIN.WMAX))-IFIX(PPDOHW(WLIN.WMIN))-1
  NPRCOL=IFIX(PPDOHW(WCOL.WMAX))-IFIX(PPDOHW(WCOL.WMIN))-1
C
C CHECK FOR DIAGNOSTICS
C
  IF(INDTOTL.NE.0) GO TO 900
C
C CLEAR SCREENS AND PRINT WINDOW HEADING
C
  CALL EAPRNT(0.1,KSON)      & ROUTE UNIT & OUTPUT TO COLOR SCREEN
  CALL EAPRNT(0.1,KSCLER)  & CLEAR COLOR SCREEN
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROPIC
004

```
WRITE(6,415) NWNDOW.MTERAL
415 FORMAT(' * WINDOW NUMBER ',J3.0X,' PICTURE',6X.4A6)
CALL EAPNT(0.1,KSOFF)      & ROUTE UNIT 6 OUTPUT TO 864 SCREEN
WRITE(6,415) NWNDOW.MTERAL
CALL IDERT( 6)
CALL IOCPRD( 6)
```

C
C
C

ANY DIAGNOSTICS???

C

```
900 IF(INDTOTL.EQ.0) GO TO 990
      IF(MDATA.C.NE.0) CALL NVIATO( PRD000.NULSUB)      & DATA/CHECKOUT
      IF(MDATA.C.EQ.0) CALL NVIATO( PRD129.PROPI9)
```

C
C
C

RETURN FOR CALL TO NAMED SUBROUTINE

C

```
990 KOHD-'
      RETURN
      END
```



```

C
C
C GLOBAL DECLARATIONS
C -----
C
      INCLUDE KOMXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
      INCLUDE KOMKLS.LIST      & COMMON CLASSIFICATION INFO
      INCLUDE KOMSYH.LIST      & COMMON SYMBOL TABLE
      INCLUDE KOMKS.LIST       & COMMON COLOR SCREEN PARAMETERS
      INCLUDE WINDEF.LIST      & DEFINE STRUCTURE OF WINDOW PACKETS
      INCLUDE KOMOMW.LIST      & COMMON OUTPUT WINDOW PACKETS
      INCLUDE KOMTBL.LIST      & COMMON TICK/FREQ TABLE
      INCLUDE PRODEF.LIST      & DEFINE PRODET PARAMETERS
      INCLUDE PXDEF.LIST       & DEFINE PIXEL BUFFER STRUCTURE
      INCLUDE MAXINT.LIST      & MAXIMUM INTEGER VALUE

C
C
C LOCAL DECLARATIONS
C -----
C
      PARAMETER NOBUFS=2      & # OF DETECTION BUFFERS IN ARRAY
      INTEGER IN DET BUF = 0 INTS PREAMBLE + (PBINS-3)/4 + #EXTRA INTS
      PARAMETER NHIDBF = 2*(PXBINS-1) + (3540-3)/4 + 27
      &#DS COLOR/INTENSITY BUF=#DS PREAM+256
      PARAMETER NHIKBF=(PXBINS-1) + 256
      PARAMETER NFRQCH=6      & # #DS PER CHANNEL IN LOCAL FREQ TABLE
      PARAMETER NFRQSZ=128    & # CHANNELS IN LOCAL FREQ TABLE
      INTEGER MOTBUF(NHIDBF,NOBUFS) & ARRAY OF DETECTION BUFFERS
      INTEGER NOBF             & DETECTION BUFFER NUMBER
      INTEGER NOBFMI          & HIGHEST DETECTION BUFFER NUMBER TO USE
      INTEGER KIBUF(NHIKBF)   & COLOR/INTENSITY BUFFER
      INTEGER NFREQ(NFRQSZ,NFRQCH) & LOCAL FREQUENCY TABLE (SCOPE INCLUDES
      & INTERNAL ROUTINE RESCOL)
      & LOCAL CROSS FREQUENCY TABLE
      INTEGER NFRQCR(10,15)
      INTEGER IPLIN           & PPD LINE
      INTEGER IPCHIN,IPCHAX   & MINIMUM AND MAXIMUM PPD COLUMN
      INTEGER IPLMIN,IPLMAX   & MINIMUM AND MAXIMUM PPD LINE
      REAL ADJLIN,ADJSAM      & ADJUSTED LINE AND SAMPLE
      INTEGER ML100L,ML100H,ML100S & MSA LINE*100: LCH,HIGH,SPACING
      INTEGER MSALIN         & MSA LINE NUMBER
      INTEGER MSASLO,MSASHI   & LOW AND HIGH MSA SAMPLE
      INTEGER ISTAT          & I/O STATUS
      INTEGER NPRLIN,NPRCOL   & NUMBER OF PPD LINES AND COLUMNS
      INTEGER LASTLN         & LAST SCAN LINE READ

C
C
C PROCEDURE
C -----
C
      CALL TRACE

C
C
C DETERMINE HOW MANY BUFFERS TO USE
C
      NOBFMI=MIN0(NLINCH,NOBUFS)
      IF(NLINCH.GT.NOBUFS) CALL MDNOTE(

```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROP13
003

= 'TOO MANY CHANNELS REQUESTED -- FIRST'.CBS4IN(NDBFMI.2).
& ' CHANNELS PROCESSED')

C
C
C

INITIALIZE MINIMUM AND MAXIMUM PPD LINES AND COLUMNS

IPLMIN=PPDOWN(HLIN.WMIN)
IPLMAX=PPDOWN(HLIN.WMAX)
IPCHIN=PPDOWN(HCOL.WMIN)
IPCMAH=PPDOWN(HCOL.WMAX)

C
C
C

COMPUTE NUMBER OF PPD LINES AND COLUMNS

NPRLIN=IPLMAX-IPLMIN+1
NPRCOL=IPCMAH-IPCHIN+1

C
C
C

CLEAR LOCAL FREQUENCY TABLES

DO 250 I=1.NLIMCH
DO 200 K=1.NFRQSZ
NFREQ(K,I)=0
200 CONTINUE
250 CONTINUE
DO 280 I=1.10
DO 270 K=1.15
NFCRO(I,K)=0
270 CONTINUE
280 CONTINUE

C
C
C

INITIALIZE LOW AND HIGH LINES AND SPACING

CALL A4P (ADJLIN.ADJSAM. FLOAT(IPLMIN).1.)
ML100L=ADJLIN*100.
CALL A4P (ADJLIN.ADJSAM. FLOAT(IPLMAX).1.)
ML100H=ADJLIN*100.
ML100S=MSAOWH(HLIN.WSP100)

C
C
C

TURN ON COLOR SCREEN

CALL EAPRNT(0.1.KSON)

C
C
C

READ. MASK. RESAMPLE. SCREEN. SYMBOLIZE AND WRITE SCAN LINES

IPLIN=IPLMIN
LASTLN=-MAXINT
DO 350 ML100=ML100L.ML100H.ML100S
MSALIN=ML100/100
CALL A4P (ADJLIN.ADJSAM. FLOAT(IPLIN).FLOAT(IPCHIN))
MSASLO=ADJSAM
CALL A4P (ADJLIN.ADJSAM. FLOAT(IPLIN).FLOAT(IPCMAH))
MSASHI=ADJSAM

```

      IF(MSALIN.EQ.LASTLN) GO TO 320      & ALREADY IN BUFFER(S)
      DO 310 NDBF=1,NDBFH
          CALL READ2N(MDTBUF(1,NDBF),(NWIDBF),ISTAT,
              MSALIN,LINCH(NDBF),MSASLO,MSASHI)
          IF((ISTAT.NE.'BADF').AND.
              (ISTAT.NE.'OFL')) GO TO 310
          CALL MDFATL( CBS4CS(ISTAT,1,4),
              ' WHILE READING DETECTION FILE')
          GO TO 900
310      CONTINUE
          LASTLN=MSALIN
320      CALL MSKPIX (MDTBUF(1,1), MDTBUF(1,1))
          IF ((MDTBUF(PXBINT,1)).EQ.'BYT').OR.
              (MDTBUF(PXBINT,1)).EQ.'CHR').OR.
              (MDTBUF(PXBINT,1)).EQ.'INT').OR.
              (MDTBUF(PXBINT,1)).EQ.'NUL') GO TO 330
          CALL MDFATL('INVALID BIN TYPE ',
              CBS4CS(MDTBUF(PXBINT,1),1,3),' IN PROPI3')
          GO TO 900
330      IF (MDTBUF(PXBINT,1)).EQ.'BYT') CALL RESCOL (KIBUF,(IPLIN),
              (IPCIN),(IPCMA), MDTBUF,(NWIDBF),(NDBUFS).GETBYT)
              &
          IF (MDTBUF(PXBINT,1)).EQ.'CHR') CALL RESCOL (KIBUF,(IPLIN),
              (IPCIN),(IPCMA), MDTBUF,(NWIDBF),(NDBUFS).GETICE)
              &
          IF (MDTBUF(PXBINT,1)).EQ.'INT') CALL RESCOL (KIBUF,(IPLIN),
              (IPCIN),(IPCMA), MDTBUF,(NWIDBF),(NDBUFS).GETINT)
              &
          IF (MDTBUF(PXBINT,1)).EQ.'NUL') CALL RESCOL (KIBUF,(IPLIN),
              (IPCIN),(IPCMA), MDTBUF,(NWIDBF),(NDBUFS).GETNUL)
              &
          CALL EAPRNT(1,N14NB(KIBUF(PXBIN)),KIBUF(PXBINS))
          IPLIN=IPLIN+1
350 CONTINUE
C
C
C MOVE DATA FROM LOCAL FREQ TABLES TO COMMON TABLE (REPLACING TICKS)
C
      KTBLTY='FREQ'
      KTBLNH=NWINDOW
      DO 800 I=1,NLIMLH
          DO 550 K=1,NFRQSZ
              KFREQ(K,1)=NFREQ(K,1)
550      CONTINUE
800 CONTINUE
      DO 700 I=1,10
          DO 650 K=1,15
              KFCRO(I,K)=NFCRO(I,K)
650      CONTINUE
700 CONTINUE
C
C
C TURN OFF COLOR SCREEN AND RETURN -- NEXT CALL IS TO PROPI9
C
900 CALL EAPRNT(0,1,KSOFF)
      CALL NVIATO ( PROI29,PROPI9)
      RETURN
C
C
C

```

C
C
C
C
C
C

INTERNAL
SUBROUTINE RESCOL (3 RESAMPLE/SCREEN/COUNT FREQUENCY/COLOR
0 KIBUF. 3 COLOR/INTENSITY BUFFER
(IPLIN. 3 PPD LINE
(IPCHIN. 3 MINIMUM PPD COLUMN
(IPCMAX. 3 MAXIMUM PPD COLUMN
"
I MOTBUF. 3 ARRAY OF MSS PIXEL BUFFERS
(NWIDBF. 3 NUMBER OF WORDS IN ONE PIXEL BUFFER
(NDBUFS. 3 NUMBER OF PIXEL BUFFERS
I GETBIN) 3 ROUTINE TO GET BIN VALUE--GETBYT.GETICE.GETINT.GETNUL

C
C
C

C METHOD

C
C
C
C
C
C
C

COMPUTE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER PREAMBLE.
FOR EACH SAMPLE. CHECK IF OUTSIDE OF INPUT WINDOW. CHECK IF OUT-
SIDE OF RADIANCE LIMITS. COUNT FREQUENCY. AND SYMBOLIZE. INSERT
TICKS.

C EXTERNAL REFERENCES

C
C
C
C
C
C

A4P 3 ADJUSTED MSS COORD FOR PRINT/PLOT COORD
GETICE 3 GET INTEGER-CHARACTER-EQUIVALENT FROM CHARACTER STRING
PUTBYT 3 PUT BYTE INTO BYTE STRING
MOVBSY 3 MOVE BYTE STRING

C GLOBAL DECLARATIONS

C
C
C
C
C
C
C
C
C

INCLUDE KOMOHM.LIST 3 COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMTBL.LIST 3 COMMON FREQ/TICK TABLE
INCLUDE KOMKLS.LIST 3 COMMON CLASSIFICATION INFO
INCLUDE PXBDEF.LIST 3 DEFINE PIXEL BUFFER STRUCTURE
INCLUDE KOMSYM.LIST 3 COMMON SYMBOL TABLE
INCLUDE KONKS.LIST 3 COMMON COLOR SCREEN PARAMETERS
INCLUDE WINDEF.LIST 3 DEFINE WINDOW PACKETS
INCLUDE MAXINT.LIST 3 DEFINE MAXIMUM INTEGER

C LOCAL DECLARATIONS

C

PARAMETER NUMBFS=6 3 NUMBER OF MPX BUFFERS
INTEGER MOTBUF(NWIDBF,NDBUFS) 3 ARGUMENT
INTEGER KIBUF(I) 3 ARGUMENT
INTEGER NBINSO(NUMBFS) 3 BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER
INTEGER MS100L,MS100H,MS100S 3 MSA SAMPLE*100: LOW,HIGH,SPACING
INTEGER MSASAM 3 MSA SAMPLE NUMBER
INTEGER KIBIN 3 COLOR/INTENSITY BUFFER BIN POINTER
REAL ADJSAM 3 ADJUSTED SAMPLE NUMBER

```

      INTEGER IPIXL1,IPIXL2,IPIXL3,
      &      IPIXL4,IPIXL5,IPIXL6      & PIXEL VALUE FOR EACH BUFFER
C
      INTEGER      IKENU      &      INTEGER-COLOR-EQUIVALENT (NEW)
      INTEGER KIKE,KIKENU      & CODED INTEGER-COLOR-EQUIVALENT (CURRENT,NEW)
      INTEGER IIE, IIENU      &      INTEGER-INTENSITY-EQUIVALENT (CURRENT,NEW)
      INTEGER KIIE      & CODED INTEGER-INTENSITY-EQUIVALENT (CURRENT)
C
C C PROCEDURE
C C
C C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
      CALL A4P (ADJLIN,ADJSAM,      FLOAT(IPLIN),FLOAT(IPCHIN))
      MS100L=ADJSAM*100.
      CALL A4P (ADJLIN,ADJSAM,      FLOAT(IPLIN),FLOAT(IPCHMAX))
      MS100H=ADJSAM*100.
      MS100S=MSAONH*(WSAM,WSPI00)
C
C C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C
      DO 150 NUMBUF=1,NDBUFS
          NBINS0(NUMBUF)=MDTBUF(PXLBIN,NUMBUF)-MDTBUF(PXLSAM,NUMBUF)
150 CONTINUE
C
C C INITIALIZE PREAMBLE FOR COLOR-INTENSITY BUFFER
C
      KIBUF(PXRECN)=MDTBUF(PXRECN,1)
      KIBUF(PXLINO)=IPLIN
      KIBUF(PXCHAN)=0
      KIBUF(PXQUAL)=0
      KIBUF(PXBINT)='BYT'
      KIBUF(PXLBIN)=1
      KIBUF(PXLCOL)=IPCHIN
      KIBUF(PXHCOL)=IPCHMAX
      KIBUF(PXNODIN)=0
      KIBUF(PXNODA)=0
      KIBUF(PXLJ01)=0
      KIBUF(PXHJ01)=0
C
C C INITIALIZE BIN POINTER & FIRST BIN OF COLOR-INTENSITY BUFFER
C
      KIBIN=KIBUF(PXLBIN)
      CALL PUTBYT(KIBUF(PXBINS),(KIBIN), 33)      & ASCII: 1
      '1' TURNS ON PICTURING IN NORTHSTAR/ISC
      <CR> <LF> AT END OF EACH LINE TURN OFF PICTURING
C
C C INITIALIZE CURRENT INTEGER-INTENSITY-EQUIV & CODED INTEGER-COLOR-EQUIV
C
      IIE=MAXINT
      KIKE=MAXINT
C

```

C
C RESAMPLE/SCREEN RADIANCE/COUNT FREQUENCY/LOOK UP SYMBOLS
C

DO 400 NS100=NS100L,NS100H,NS100S
NSASAM=NS100/100

C
C BUFFER 1
C

IF ((NSASAM.LT.MDTBUF(PXLSAM.1)).OR.
(NSASAM.GT.MDTBUF(PXHSAM.1))) GO TO 350 & SAMPLE NOT IN BUFFER
CALL GETBIN (IPIXL1,
 MDTBUF(PXBINS.1).(NSASAM*NBINS0(1)))
IPIXL2=IPIXL1 & SECOND CHANNEL VALUE SAME AS FIRST IF UNDEFINED
IF ((IPIXL1.GE.MDTBUF(PXNODA.1)) GO TO 350 & NO DATA
IF ((IPIXL1.LT.LCVLO1).OR.
(IPIXL1.GT.LCVHI1)) GO TO 360 & OUT OF RAD LIMITS
IF (NLINCH-1.EQ.0) GO TO 320

C
C BUFFER 2
C

IF ((NSASAM.LT.MDTBUF(PXLSAM.2)).OR.
(NSASAM.GT.MDTBUF(PXHSAM.2))) GO TO 350 & SAMPLE NOT IN BUFFER
CALL GETBIN (IPIXL2,
 MDTBUF(PXBINS.2).(NSASAM*NBINS0(2)))
IF ((IPIXL2.LT.LCVLO(2)).OR.
(IPIXL2.GT.LCVHI(2))) GO TO 360 & OUT OF RAD LIMITS
IF (NLINCH-2.EQ.0) GO TO 290

C
C BUFFER 3
C

IF ((NSASAM.LT.MDTBUF(PXLSAM.3)).OR.
(NSASAM.GT.MDTBUF(PXHSAM.3))) GO TO 350 & SAMPLE NOT IN BUFFER
CALL GETBIN (IPIXL3,
 MDTBUF(PXBINS.3).(NSASAM*NBINS0(3)))
IF ((IPIXL3.LT.LCVLO(3)).OR.
(IPIXL3.GT.LCVHI(3))) GO TO 360 & OUT OF RAD LIMITS
IF (NLINCH-3.EQ.0) GO TO 260

C
C BUFFER 4
C

IF ((NSASAM.LT.MDTBUF(PXLSAM.4)).OR.
(NSASAM.GT.MDTBUF(PXHSAM.4))) GO TO 350 & SAMPLE NOT IN BUFFER
CALL GETBIN (IPIXL4,
 MDTBUF(PXBINS.4).(NSASAM*NBINS0(4)))
IF ((IPIXL4.LT.LCVLO(4)).OR.
(IPIXL4.GT.LCVHI(4))) GO TO 360 & OUT OF RAD LIMITS
IF (NLINCH-4.EQ.0) GO TO 230

C
C BUFFER 5
C

IF ((NSASAM.LT.MDTBUF(PXLSAM.5)).OR.
(NSASAM.GT.MDTBUF(PXHSAM.5))) GO TO 350 & SAMPLE NOT IN BUFFER
CALL GETBIN (IPIXL5,
 MDTBUF(PXBINS.5).(NSASAM*NBINS0(5)))
IF ((IPIXL5.LT.LCVLO(5)).OR.
(IPIXL5.GT.LCVHI(5))) GO TO 360 & OUT OF RAD LIMITS

IF (NLINCH-S.EQ.0) GO TO 200

C
 C BUFFER 6
 C

IF ((MSASAM.LT.MDTBUF(PXLSAM,6)).OR.
 (MSASAM.GT.MDTBUF(PXMSAM,6))) GO TO 350 6 SAMPLE NOT IN BUFFER
 CALL GETBIN (PIXLS,
 MDTBUF(PXBINS,6),(MSASAM+NBINS0(6)))
 IF ((PIXLS.LT.LCVLO(6)).OR.
 (PIXLS.GT.LCVHI(6))) GO TO 360 6 OUT OF RAD LIMITS

C
 C COUNT FREQUENCY
 C

200 NFREQ(PIXLS+1,6)=NFREQ(PIXLS+1,6)+1
 230 NFREQ(PIXLS+1,5)=NFREQ(PIXLS+1,5)+1
 260 NFREQ(PIXLS+1,4)=NFREQ(PIXLS+1,4)+1
 290 NFREQ(PIXLS+1,3)=NFREQ(PIXLS+1,3)+1
 320 NFREQ(PIXLS+1,2)=NFREQ(PIXLS+1,2)+1
 NFREQ(PIXLS+1,1)=NFREQ(PIXLS+1,1)+1

C
 C
 C INFO -- LOOK UP NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
 C

CALL GETICE(IIEU,
 KSYM(PIXL1+1),(5))
 CALL GETICE(IKENU,
 KSYM(PIXL2+1),(6))
 GO TO 370

C
 C
 C NO DATA -- ASSIGN NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
 C

350 CALL GETICE(IIEU,
 KSYM(ISYND+1),(5))
 CALL GETICE(IKENU,
 KSYM(ISYND+1),(6))
 GO TO 370

C
 C
 C NO INFO -- ASSIGN NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
 C

380 CALL GETICE(IIEU,
 KSYM(ISYMI+1),(5))
 CALL GETICE(IKENU,
 KSYM(ISYMI+1),(6))

C
 C
 C COUNT INTENSITY X COLOR CROSS FREQUENCY
 C

370 NFCRO(IIEU+1,IKENU+1)=NFCRO(IIEU+1,IKENU+1)+1

C
 C
 C IF NEW CODED I-K-E. PUT CODED I-K-E & CODED I-I-E IN BUFFER
 C

KIKENU=KSIKE(IKENU+1)
 IF(IKIKENU.EQ.KIKE) GO TO 360

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROPI3
009

```

      KIKE=KIKENU
      KIBIN=KIBIN+1
      CALL PUTBYT(KIBUF(PXBINS).(KIBIN). KIKE)
      IIE=IIENU
      KIIE=IIE+97
      KIBIN=KIBIN+1
      CALL PUTBYT(KIBUF(PXBINS).(KIBIN). KIIE)
      GO TO 400

C
C
C SAME COLOR -- IF SAME INTENSITY & BIN. UPDATE CODED IIE IN BUFFER
C
380   IF(IIENU.NE.IIE) GO TO 390
      IF(KIIE.GT.116) GO TO 390
      KIIE=KIIE+10
      CALL PUTBYT((KIBUF(PXBINS).KIBIN). KIIE)
      GO TO 400

C
C
C SAME COLOR WITH NEW INTENSITY OR NEW BIN
C
390   IIE=IIENU
      KIIE=IIE+97
      KIBIN=KIBIN+1
      CALL PUTBYT(KIBUF(PXBINS).(KIBIN). KIIE)

C
C
C LOOP TO PROCESS NEXT PIXEL
C
400 CONTINUE

C
C
C STORE POINTER TO LAST BIN IN PREAMBLE & PAD BUFFER WITH 0 NUL'S
C
      KIBUF(PXMBIN)=KIBIN
      CALL MOVBST(KIBUF(PXBINS).(KIBIN+1).(8).
      . 0.(1).(1).0)

C
C
900 RETURN
      END
```


ALL NVIATO(PRO000,NULSUB)

DIAGNOSTICS???

```
IF(INDFATL.EQ.0) GO TO 920
  CALL MDNOTE( 'FATAL ERRORS -- NO PICTURE GENERATED')
  GO TO 990
IF(INDWARN.EQ.0) GO TO 980
  CALL MDNOTE( 'PREVIOUS WARNINGS -- NO PICTURE GENERATED')
  IF(INDATCH.EQ.0) WRITE(6,925)
  FORMAT(4X,'**TRY AGAIN!')
  CALL MDCLR( NULCST)
  GO TO 990
```

PRINT MSA OUTPUT WINDOW COORDINATES AND PREPARE FOR NEXT WINDOW

```
160 WRITE(6,965) MSAOWH(HLIN,WHIN)
165 FORMAT(1X,11X,'LINE ',14)
  WRITE(6,975) MSAOWH(MSAM,WHIN),MSAOWH(MSAM,WHAX)
175 FORMAT(1X,'SAMPLE ',14,9X,'SAMPLE ',14)
  WRITE(6,985) MSAOWH(HLIN,WHAX)
  WRITE(6,985)
185 FORMAT(1X)
  NWINDOW=NWINDOW+1
```

C RETURN TO NEXT STATEMENT IN CALLING ROUTINE

990 RETURN
END

ORIGINAL PAGE IS
OF POOR QUALITY

SUBROUTINE PROXQT 8 INITIALIZATION ROUTINE FOR PRDDET

```
C
C
C
C HISTORY
C -----
C
C      E M SCHLOSSER   LEC      11/29/75   ORIGINAL CODE
C      E M SCHLOSSER   LEC      01/23/79   ALLOW DEFAULT COMMANDS FROM MACDAM
C      J C CRISP       LEC      12/21/79   INITIALIZE NLMCH & DELETE MINC
C      J C CRISP       LMSCO    05/16/80   LINCH(1)=1. CHG PRDENS TO PRDDET
C
C
C METHOD
C -----
C
C      INITIALIZE PROGRAM. OPEN FILES. IDENTIFY SCENE. QUEUE DEFAULT COMMANDS.
C
C MACHINE-DEPENDENT CODE
C -----
C
C      UNIVAC EXEC-8 PROGRAM FILE NAMING CONVENTIONS.
C
C
C EXTERNAL REFERENCES
C -----
C
C      NVIATO      8 NAME NEXT 'VIA' & 'TO' ROUTINES
C      PSTART     8 PROGRAM START INITIALIZATION
C      SYSADD     8 ADD DISK SYMBOLIC FILE OR ELT TO SYSDIN RUNSTREAM
C      OPNIZN     8 OPEN INPUT DETECTION FILES (21. 22. ... ..)
C      CLSHDD     8 PRINT CLASSIFICATION HEADING
C      HOFATL     8 SUBMIT FATAL DIAONOSTIC MESSAGE
C      EXTERNAL  PRODD0.  NULSUB
C
C
C EXCEPTIONS
C -----
C
C      1. MISSING DEFAULT COMMANDS GENERATE A FATAL DIAONOSTIC.
C
C
C GLOBAL DECLARATIONS
C -----
C
C      INCLUDE KOMXQT.LIST      8 COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
C      INCLUDE KOMKLS.LIST     8 COMMON CLASSIFICATION INFO
C
C
C LOCAL DECLARATIONS
C -----
C
C      INTEGER LOCFIL      8 LOCATION WITHIN DISK SYMBOLIC FILE ( IF > 0 )
C
C
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROGOT
00B

C PROCEDURE

C -----

C

C

C IDENTIFY PROGRAM

C

CALL PSTART('DAM PRYDET(0009)')

C

C

C ON RETURN, CALL PRD000 TO GET DEFAULT/USER COMMANDS

C

CALL NVIATO(PRD000.NULSUB)

C

C

C OPEN DETECTION FILES AND IDENTIFY ERTS SCENE

C

CALL OPN12N

CALL CLSHOO(6)

C

C

C INITIALIZE LIMIT CHANNEL TO FIRST DETECTION CHANNEL

C

NLIMCH=1

LIMCH(1)=1

C

C

C QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE

C

CALL SYSADDILOCFIL. 'MACDAM','DEF-PRYDET',' '

IF(LOCFIL.LE.0) CALL SYSADDILOCFIL. 'DAM','DEF-PRYDET',' '

IF(LOCFIL.LE.0) CALL MDFATL('NO DEFAULT COMMANDS')

C

C

C RETURN TO NEXT STATEMENT IN CALLING ROUTINE

C

RETURN

END


```

INCLUDE KOKNLS.LIST      & COMMON CLASSIFICATION INFO
INCLUDE KONSYM.LIST     & COMMON SYMBOL TABLE
INCLUDE MINDEF.LIST    & DEFINE TWO-DIMENSIONAL WINDOW
INCLUDE KONOHW.LIST    & DEFINE OUTPUT WINDOW PACKET
INCLUDE ICBUFI.LIST    & DECLARE CHARACTER BUFFER
  
```

```

C
C LOCAL DECLARATIONS
C -----
  
```

```

C      INTEGER NAMFIL (2)      & ALTERNATE PRINT FILE NAME
  
```

```

C
C PROCEDURE
C -----
  
```

```

C      CALL TRACE
  
```

```

C
C INITIALIZE FILE NAME AND CHARACTER BUFFER
  
```

```

C      CALL CSTRIN (NAMFIL(1),122), (UNIT,1)
  
```

```

C
C CHECK FOR VALID OUTPUT UNIT/PRINT FILE
  
```

```

C      IF ((UNIT.EQ.8).OR.
& ((UNIT.EQ.10).AND.(UNIT.LE.(10-MATH-1)))) GO TO 400
C      CALL MCFATL (CSTRIN(UNIT,1)),' ','IS BAD UNIT IN 10CPRO'
C      GO TO 900
  
```

```

C
C IDENTIFY CHANNEL NUMBER (1)
  
```

```

C 400 CALL CBINIT (ICBUFI)
C      CALL CB4CST (ICBUFI, 'CHAN.0ET')
C      DO 450 NLC=1,NLIMCH
C          CALL CB4CST (ICBUFI, ' ')
C          CALL CB4IN (ICBUFI, LIMCH/NLC,1)
C 450 CONTINUE
  
```

```

C
C IDENTIFY RADIANCE LIMIT VALUES AND SYMBOLS FOR FIRST CHANNEL
  
```

```

C      CALL CB4CST (ICBUFI, '...')
C      CALL CB4CST (ICBUFI, KLSTYP,(1),(3))
C      CALL CB4CST (ICBUFI, ' ')
C      CALL CB4IN (ICBUFI, LCVLO1,1)
C      CALL CB4CST (ICBUFI, ' ')
C      CALL CB4CST (ICBUFI, KSYM(LCVLO1-1),(1),(1))
C      CALL CB4CST (ICBUFI, ' ')
C      CALL CB4IN (ICBUFI, LCVH1,1)
C      CALL CB4CST (ICBUFI, ' ')
C      CALL CB4CST (ICBUFI, KSYM(LCVH1-1),(1),(1))
  
```

```

C
C
  
```

```
C IDENTIFY RADIANCE LIMIT VALUES FOR ANY OTHER CHANNELS
C
  IF (NLINCH.LT.4) GO TO 600
  DO 550 NLC=2,NLINCH
    CALL CB4CST (ICBUF1, '...', '...')
    CALL CB4IN (ICBUF1, '...', LCVLO(NLC),1)
    CALL CB4CST (ICBUF1, '...', '...')
    CALL CB4IN (ICBUF1, '...', LCVHI(NLC),1)
  550 CONTINUE
C
C
C IDENTIFY SPACING
C
  600 CALL CB4CST (ICBUF1, '...', '...SPA')
  DO 650 NAXIS=1,2
    CALL CB4CST (ICBUF1, '...', '...')
    SPA=FLOAT(MSAOHM(NAXIS,WSP100))/100.
    IF (SPA.EQ.AINT(SPA)) CALL CB4IN (ICBUF1, '...', IFIX(SPA),1)
    IF (SPA.NE.AINT(SPA)) CALL CB4RL (ICBUF1, '...', SPA,1,4)
  650 CONTINUE
C
C
C IDENTIFY ORIGIN
C
  CALL CB4CST (ICBUF1, '...', '...ORIG.SCAN.')
  CALL CB4IN (ICBUF1, '...', MSAOHM(HLIN,WORIG),1)
  CALL CB4CST (ICBUF1, '...', '...')
  CALL CB4IN (ICBUF1, '...', MSAOHM(WSAM,WORIG),1)
  CALL CB4CST (ICBUF1, '...', '...')
C
C
C OUTPUT CHANNEL/RADIANCE/SPACING/ORIGIN INFO
C
  IF (IUNIT.EQ.6) CALL ERPRNT (1,22,ICBUF1)
  IF (IUNIT.NE.6) CALL ERPRTA (NAMFIL,1,22,ICBUF1)
C
C
C NORMAL RETURN
C
  900 RETURN
  END
```


DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

OPRPRD
002

```
C -----  
C  
C      CALL TRACE  
C  
C  
C NO MORE THAN HALTHI ALTERNATE PRINT FILES FOR PRTOET  
C  
C      HALTM=MAX0(HALTM.1)  
C      HALTM=MIND(HALTM.HALTHI)  
C  
C  
C NO MORE THAN KPAGHI COLUMNS PER PAGE FOR PRTOET  
C  
C      KPAGE=MAX0(KPAGE.1)  
C      KPAGE=MIND(KPAGE.KPAGHI)  
C  
C  
C OPEN FILE(S)  
C  
C      IF(MDATAC.NE.0) GO TO 900      @ DATA/CHECKOUT MODE  
C      CALL OPENPR  
C  
C  
C IDENTIFY ERTS SCENE  
C  
C      WRITE(10.225)  
C      225 FORMAT(1X)      @ SKIP LINE  
C      CALL CLSHOG(10)  
C  
C  
C 900 RETURN  
C      END
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRTCLASS
001

C PROGRAM PRTCLASS
C -----
C

C HISTORY
C -----
C

C E H SCHLOSSER LEC 07/02/73 ORIGINAL CODE
C MARY TOMPKINS LEC 01/05/80 UPGRADE DOCUMENTATION
C

C METHOD
C -----
C

C THIS PROGRAM PRODUCES MAPS OF CLASSIFIED ERTS MSS DATA ON A COMPUTER
C LINE PRINTER. THE USER SPECIFIES THE SCALE, WINDOW DIMENSIONS,
C LOCATION, ETC. OF EACH MAP EACH MAP.
C

C MACHINE-DEPENDENT CODE
C -----
C

C NONE.
C

C EXTERNAL REFERENCES
C -----
C

C NVIATO & NAME 'VIA' 'TO' ROUTINES
C VIATO & CALL 'VIA' 'TO' ROUTINES
C VIA TO
C EXTERNAL PRC000. PRCXQT
C

C EXCEPTIONS
C -----
C

- C 1. THIS PROGRAM IS LIMITED TO THAT PART OF THE WORLD COVERED
C BY THE CLARKE 1866 SPHEROID (NORTH AMERICA).
C
C 2. THE PROGRAM CLASSIFY MUST BE EXECUTED BEFORE THIS PROGRAM.
C PREFERABLY IN THE SAME RUN.
C
C 3. IF PRCXQT DOES NOT CALL NVIATO TO CHANGE THE 'VIA' AND/OR 'TO'
C ROUTINES. THEN PRTCLASS WILL CALL TO PRCXQT IN AN ENDLESS LOOP!
C

C GLOBAL DECLARATIONS
C -----
C

INCLUDE KOMXQT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONLOG.LIST & COMMON LOG FILE BUFFER, I/O PKT. POINTERS
INCLUDE KOMLUS.LIST & COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
INCLUDE KOML2N.LIST & COMMON I/O PKTS FOR DETECTION FILES (UNITS 2N)
INCLUDE KOMIHW.LIST & COMMON INPUT WINDOW PACKETS
INCLUDE KOMOHV.LIST & COMMON OUTPUT WINDOW PACKETS

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRTCLASS
002

INCLUDE KOMNER.LIST	8 COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST	8 COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST	8 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMTBL.LIST	8 COMMON BLOCKS AND DEFINE PROCEDURES
INCLUDE KOMDET.LIST	8 COMMON DETECTION FILE WINDOW PACKETS & DATES
INCLUDE KOMSYH.LIST	8 COMMON SYMBOL TABLE
INCLUDE KOMALT.LIST	8 COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS

C
C
C
C
C

PROCEDURE

CALL NVIATO(PRC000.PRCXQT) 8 FIRST CALL IS VIA PRC000 TO PRCXQT
100 CONTINUE
 CALL VIATO
 GO TO 100
END 8 (STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE)

PROGRAM PRTCLASS/VIRTUAL

HISTORY

E H SCHLOSSER	LEC	08/02/74	ORIGINAL CODE
E H SCHLOSSER	LEC	11/08/79	MAP.FZINI: NO 'N' IN DEMAND

METHOD

CONSTRUCT MAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT BXQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE MAP & BXQT COMMANDS TO TEMPORARY FILE 20.
ADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 8-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES.
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS & FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IOWS & INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS & TERMINATE PROGRAM EXECUTION
DAM.PRTCLASS-MAP & SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAOPT & STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS BASO-D & PREP-D.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

(2 = REAL TIME
(3 = LOW EXEC
(4 = DEMAND
(5 = DEADLINE BATCH
(6 = BATCH

(BXQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER A5 IN

(MASTER BIT NOTATION.

LOCAL DECLARATIONS

```

      AXRS
S(00) . D-BANK
SSSH   FORM      6.6.6.18
      111111222222333333444444555555666666777777888888999999
LABSDF  SSSH      050.1.'F'.0 . LABEL. 1 WD. FORTRAN. FIELDATA
MAPSDF  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
MAPING  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
ADDSDf  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
ADDING  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
XQTSDF  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
XQTING  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
EOfSDF  -         0 . END-OF-FILE STOP WORD
PF      FORM      12.6.18
CSFASG  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
CSFADD  SSSH      000.9.0.0 . DATA. 9 WDS. . FIELDATA
SAVREG  RES       1
IOPKT   RES       1
      '20'.WS 33.LABSDF.'0' 0
  
```

PROCEDURE

```

S(01) . I-BANK
PRTCLASS LA.U      A0.' ' . A0 := ' '
          TNE.U     A4.4 . SKIP NEXT INST IF A4<>4 (NOT DEMAND)
          SA.S2     A0.MAPING+2 . DEMAND! BLANK OUT N OPTION
          LA        A0.(CSFASG) . ADDRESS OF 8ASG IMAGE
          ER        CSFS . DO IT
          SA        A0.SAVREG . STORE 8
          PPRINT    (PF 2.1.SAVREG) . PRINT 8ASG STATUS
          GETOPT    . LOAD OPT LTRS INTO A2.A3.A4
          PUTOPT    DS      A2.XQTING+2 . STORE OPTION LETTERS INTO 8XQT IMAGE
          SA        A4.XQTING+4 . (3 WORDS -- MAX 18 OPT LETTERS)
          WRITE     LA      A0.(IOPKT) . ADDRESS OF I/O PACKET
          ER        IOWS . WRITE SDF IMAGES TO 20.
          ADD       LA      A0.(CSFADD) . ADDRESS OF 8ADD IMAGE
          ER        CSFS . DO IT
          ER        EXITS
          END       PRTCLASS
  
```

PRTCLASS OVERLAY STRUCTURE

HISTORY

E H SCHLOSSER	LEC	03/27/75	ORIGINAL CODE
E H SCHLOSSER	LEC	07/14/78	CHANGE OVERLAYS TO REDUCE THRASHING
E H SCHLOSSER	LEC	01/31/79	MACRO COMMANDS & TIME COMMAND
MARY TOMPKINS	LEMSCO	01/16/80	PEEK,POKE,IF,FI & OPTIMIZE OVERLAYS
MARY TOMPKINS	LEMSCJ	05/16/80	CHANNEL COMMAND

LID DAM.

SEG S-MAIN

IN DAM.PRTCLASS/ . MAIN PROGRAM
IN DAM.NVIATO . NAME/CALL 'VIA' AND 'TO' SUBROUTINES
IN DAM.NULSUB . DO NOTHING
IN DAM.SYS-BLOCK . BLOCK DATA SUBROUTINE

. MONITOR FOR PHASE 0 COMMANDS -----

IN DAM.PRC000 . CALL USER-SPECIFIED PHASE 0 ROUTINE
IN DAM.NTABS/DAM . DAM UNIT * TABLE GOES IN SAME SEG W/ FORTRAN I/O

. UTILITIES FOR PHASE 0 COMMANDS -----

SEG S-READS*(S-MAIN)

IN DAM.READS . 'READ' INTO UNIT 5 BUFFER
IN DAM.GETS . 'GET' FREE-FORMAT FIELD FROM UNIT 5 BUFFER
IN DAM.WARNS . PROCESS WARNING DIAGNOSTIC FOR UNIT 5 FIELD

SEG S-OPNCLPR*(S-MAIN)

IN DAM.OPRPRC . OPEN ALT PRT FILE
IN DAM.CLOSPR . CLOSE ALT PRT FILE

SEG S-DUMP*(S-MAIN)

IN DAM.DMPTIC . DUMP TIC TABLE
IN DAM.DMPWIN . DUMP WINDOW PACKETS

. PHASE 0 COMMANDS (FORTRAN I/O ALLOWED) -----

SEG START-STOP*(S-READS,S-OPNCLPR,S-DUMP)

IN DAM.PRCXQT . PRTCLASS INITIALIZATION ROUTINE
IN DAM.PRCXET . PRTCLASS TERMINATION ROUTINE

SEG SPECIFY*.START-STOP

IN DAM.DETCHA . GET/CHECK DETECTION CHANNEL NUMBER(S)
IN DAM.KMDCLE . CLEAR WARNINGS/ERRORS
IN DAM.KMDCOP . GET/CHECK NUMBER OF OUTPUT COPIES
IN DAM.KMDCOU . GET/CHECK COUNT PER PIXEL

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRTCLASS-MAP
002

IN DAN.KNDEN . GET/CHECK DENSITY LIMITS
IN DAN.KNDXP . EXPLAIN PROGRAM/COMMAND
IN DAN.KNDFI . END IF...FI BLOCK
IN DAN.KNDHEA . GET/CHECK PAGE HEADING(S)
IN DAN.KNDIF . BEGIN IF...FI BLOCK
IN DAN.KNDMER . GET/CHECK TRANSVERSE MERCATOR CENTRAL MERIDIAN
IN DAN.KNDNEW . PRINT NEWS
IN DAN.KNDNEX . CONDITIONALLY PERFORM NEXT COMMAND
IN DAN.KNDOFF . TURN OFF MODE SWITCH(ES)
IN DAN.KNDON . TURN ON MODE SWITCH(ES)
IN DAN.KNDORI . GET/CHECK WINDOW ORIGIN
IN DAN.KNDPAG . SKIP TO TOP OF NEXT PAGE
IN DAN.KNDPEE . PEEK
IN DAN.KNDPOK . POKE
IN DAN.KNDPRI . GET/CHECK PRINTER SPECIFICATIONS
IN DAN.KNDRAD . GET/CHECK RADIANCE LIMITS
IN DAN.KNDREN . RENUMBER (GET/CHECK NEW WINDOW SEQUENCE NUMBER)
IN DAN.KNDSCA . GET/CHECK WINDOW SCALE
IN DAN.KNDSYM . GET/CHECK SYMBOLS
IN DAN.KNDTIC . GET/CHECK TICK UNITS/INTERVALS
IN DAN.KNDTIM . PRINT CLOCK TIME & CHARGE TIME
IN DAN.KNDWIN . GET/CHECK WINDOW ENVELOPE/VERTICES
IN DAN.KNDXXX . MACRO COMMANDS
IN DAN.KNDZON . GET/CHECK UTM PROJECTION ZONE
IN DAN.KNDZAD . DYNAMIC ZADD
IN DAN.KNDZAS . DYNAMIC ZASO
IN DAN.KNDZBR . DYNAMIC ZBRKPT
IN DAN.KNDZFR . DYNAMIC ZFREE
IN DAN.KNDZLO . DYNAMIC ZLOO

SEO MAPOUT*.START-STOP

IN DAN.PRCHAP . MAP RADIANCE/DENSITY/CLASS (PHASE 0)

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRCLASS-NAP/VIRTUAL
001

IN DAN.PRCLASS/VIRTUAL


```

C
C   INTEGER KOMD           & FIRST 3 CHARS OF USER COMMAND (BLANK AFTER DONE)
C   INTEGER LSSTAT        & READS STATUS ('EOF' MEANS END-OF-FILE)
C   INTEGER KASE          & MODIFIED I-C-E OF FIRST CHAR OF COMMAND
C
C C
C C PROCEDURE
C C -----
C C
C C CALL PREVIOUSLY NAMED SUBROUTINE
C C
C C   CALL TRACE
C C   CALL NANSUB           & CALL TO NULSUB DOES NOTHING
C C
C C
C C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C C
C C   KOMD=' NUL'           & IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
C C   CALL READS(LSSTAT, ' ') & FILL BUFFER. BLANK CUE MESSAGE
C C   IF(LSSTAT.NE.' ') KOMD='EOF5'
C C   IF(KOMD.NE.'EOF5') CALL GETSAL(KOMD(1), NULCST) & GET 3 ALPHA CHARS
C C
C C
C C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C C
C C   KASE=ICE(KOMD)-ICE('A')+1 & A TO Z = 1 TO 26
C C
C C
C C CASE STATEMENT ON MODIFIED I-C-E OF COMMAND'S FIRST CHARACTER
C C
C C   IF((KASE.LT.1).OR.(KASE.GT.26)) KASE=27 & NOT ALPHA
C C   GO TO 1
C C   0 401.402.403.404.405.406.407.408.409.410.
C C   1 411.412.413.414.415.416.417.418.419.420.
C C   2 421.422.423.424.425.426.427)
C C   & .KASE
C C
C C
C C DETERMINE COMMAND. PERFORM COMMAND. CHANGE KOMD TO BLANK
C C
C C   401 CONTINUE &... A
C C   402 CONTINUE &... B
C C   GO TO 800
C C
C C
C C   403 CONTINUE &... C
C C   IF(KOMD.EQ.'CHA') CALL DETCHA(KOMD) & CHANNEL (DETECTION)
C C   IF(KOMD.EQ.'CLE') CALL KNDCLL(KOMD) & CLEAR
C C   IF(KOMD.EQ.'COP') CALL KNDCOPI(KOMD) & COPIES
C C   IF(KOMD.EQ.'COU') CALL KNDCOU(KOMD) & COUNT
C C   GO TO 800
C C
C C
C C   404 CONTINUE &... D
C C   IF(KOMD.EQ.'DEN') CALL KNDDEN(KOMD) & DENSITY
C C   GO TO 800
C C
C C
  
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRC000
003

405 CONTINUE 8000 E
IF(KOND.EQ.'EOF8') CALL PRCEX1(KOND) 8 END-OF-FILE CAUSES EXIT
IF(KOND.EQ.'EX1') CALL PRCEX1(KOND) 8 EXIT
IF(KOND.EQ.'EXP') CALL KNDEXP(KOND) 8 EXPLAIN
GO TO 800

C

406 CONTINUE 8000 F
IF(KOND.EQ.'FI ') CALL KNDFI (KOND) 8 FI (ENDIF)
GO TO 800

C

407 CONTINUE 8000 G
GO TO 800

C

408 CONTINUE 8000 H
IF(KOND.EQ.'HEA') CALL KNDHEA(KOND) 8 HEADING
GO TO 800

C

409 CONTINUE 8000 I
IF(KOND.EQ.'IF ') CALL KNDIF (KOND) 8 IF
GO TO 800

C

410 CONTINUE 8000 J
411 CONTINUE 8000 K
412 CONTINUE 8000 L
GO TO 800

C

413 CONTINUE 8000 M
IF(KOND.EQ.'MAP') CALL PRCHAP(KOND) 8 MAP
IF(KOND.EQ.'MER') CALL KNDMER(KOND) 8 MERIDIAN
GO TO 800

C

414 CONTINUE 8000 N
IF(KOND.EQ.'NEW') CALL KNDNEW(KOND) 8 NEWS
IF(KOND.EQ.'NEX') CALL KNDNEX(KOND) 8 NEXT
GO TO 800

C

415 CONTINUE 8000 O
IF(KOND.EQ.'OFF') CALL KNDOFF(KOND) 8 OFF
IF(KOND.EQ.'ON ') CALL KNDON (KOND) 8 ON
IF(KOND.EQ.'ORI') CALL KNDORI(KOND) 8 ORIGIN
GO TO 800

C

416 CONTINUE 8000 P
IF(KOND.EQ.'PA0') CALL KNDPA0(KOND) 8 PAGE
IF(KOND.EQ.'PEE') CALL KNDPEE(KOND) 8 PEEK
IF(KOND.EQ.'POK') CALL KNDPOK(KOND) 8 POKE
IF(KOND.EQ.'PRI') CALL KNDPRI(KOND) 8 PRINT
GO TO 800

C

417 CONTINUE 8000 Q
GO TO 800

C

418 CONTINUE 8000 R
IF(KOND.EQ.'RAD') CALL KNDRAD(KOND) 8 RADIANCE
IF(KOND.EQ.'REN') CALL KNDREN(KOND) 8 RENUMBER
GO TO 800

DAN PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

PRC000
 004

```

C
C 419 CONTINUE 3*** S
      IF(KOMD.EQ.'SCA') CALL KHDSCA(KOMD)      3 SCALE
      IF(KOMD.EQ.'SYN') CALL KHD SYN(KOMD)     3 SYMBOLS
      GO TO 800
C
C 420 CONTINUE 3*** T
      IF(KOMD.EQ.'TIC') CALL KHD TIC(KOMD)     3 TICKS
      IF(KOMD.EQ.'TIM') CALL KHD TIM(KOMD)     3 TIME
      GO TO 800
C
C 421 CONTINUE 3*** U
C 422 CONTINUE 3*** V
      GO TO 800
C
C 423 CONTINUE 3*** W
      IF(KOMD.EQ.'WIN') CALL KHD WIN(KOMD)     3 WINDOW
      GO TO 800
C
C 424 CONTINUE 3*** X
C 425 CONTINUE 3*** Y
      GO TO 800
C
C 426 CONTINUE 3*** Z
      IF(KOMD.EQ.'ZON') CALL KHD ZON(KOMD)     3 ZONE
      GO TO 800
C
C 427 CONTINUE 3*** NOT ALPHABETIC
      IF(KOMD.EQ.'SAD') CALL KHD SAD(KOMD)     3 SADD
      IF(KOMD.EQ.'SAD') CALL KHD SAD(KOMD)     3 SADD
      IF(KOMD.EQ.'SAS') CALL KHD SAS(KOMD)     3 SASO
      IF(KOMD.EQ.'SBR') CALL KHD SBR(KOMD)     3 SBRKPT
      IF(KOMD.EQ.'SFR') CALL KHD SFR(KOMD)     3 SFREE
      IF(KOMD.EQ.'SLO') CALL KHD SLO(KOMD)     3 SLOG
C
C
C IF COMMAND WAS NOT FOUND. TRY MACRO-COMMAND
C
C 800 IF(KOMD.NE.' ') KOMD='PRC-'      3 1ST 3 CHARS OF PROG NAME PLUS '--'
      IF(KOMD.NE.' ') CALL KHD XXX(KOMD)      3 MACRO COMMAND HANDLER
C
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
      IF(KOMD.NE.' ') CALL WARN5('INVALID COMMAND --')
C
C
C FORCE ALL FORTRAN I/O ROUTINES INTO SAME SEG AS PRC000 (NEVER PERFORMED)
C
      IF(KOMD.EQ.'JUNK') READ(895,895) KOMD
      895 FORMAT(1X)
C
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
      RETURN
  
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRC000
005

END

SUBROUTINE PRCEX1(0 TERMINATION ROUTINE FOR PRCLASS
U KOND) 0 1: FIRST 3 CHARS OF COMMAND 0: SPACES

C
C
C (E H SCHLOSSER)
C
C
C INCLUDE KONXQT.LIST
C INCLUDE NULCST.LIST 0 DEFINE NULL CHARACTER STRING
C CALL TRACE
C
C WRITE(6,125)
125 FORMAT(4X,'**PROGRAM TERMINATION')
IF(INDATAC.EQ.0) CALL CLOSPR
IF(INDFATL.NE.0) CALL PABORT(NULCST)
WRITE(6,165)
165 FORMAT(' DO YOU WANT DETECTION FILES SAVED?')
CALL READ5(L5STAT, ' ')
IFSAVE='N'
CALL GETSKH(IFSAVE.(1), NULCST)
C
C
C DELETE DETECTION FILES
C
C IF(IFSAVE.NE.'Y') CALL DLET2N
C
C
C TERMINATE PROGRAM
C
C 900 CALL PSTOP(NULCST)
C
C
C PSTOP DOES NOT RETURN
C
C
C END


```

      INCLUDE WINDEF.LIST      & DEFINE STRUCTURE OF WINDOW PACKETS
      INCLUDE KOMINH.LIST     & COMMON INPUT WINDOW PACKETS
      INCLUDE KOMOHV.LIST     & COMMON OUTPUT WINDOW PACKETS
      INCLUDE NULCST.LIST     & DEFINE NULL CHARACTER STRING

C
C
C LOCAL DECLARATIONS
C -----
C
      INTEGER NSUBW           & NUMBER OF SUB-WINDOWS REQUESTED
      INTEGER INTEMP         & TEMPORARY

C
C
C PROCEDURE
C -----
C
      CALL TRACE

C
C
C GET/CHECK NUMBER OF SUB-WINDOWS
C
      NSUBW=0
      CALL GETSIN(NSUBW, 1,500,'BAD NUMBER OF SUB-WINDOWS --')
      IF(MCFIRM.NE.0) WRITE(6,115) NSUBW
115 FORMAT(' MAP, '.13,' SUB-WINDOWS')

C
C
C CALIBRATE/CHECK SPECIFICATIONS
C
      IF(NWINDOW.EQ.0) CALL WARN5('INVALID DEFAULT COMMAND --')
      CALL GETSIN(INTEMP, +1,-1,'EXTRA MAP SPECIFICATION --')
      IF(NLIMCH.GT.1) CALL MDNOTE(
- 'MORE THAN 1 CHANNEL REQUESTED -- ONLY FIRST CHANNEL PROCESSED')
      CALL CALSYM
      CALL CALSCA( FLOAT(LINCH),FLOAT(KINCH))
      CALL CALWIN( 2.)
      IF(MCHECK.NE.0) GO TO 900

C
C
C CHECK IF ANY DATA LIES WITHIN OUTPUT WINDOW
C
      IF((MSAOWH(WLIN,WMIN).GE.MSAIWH(WLIN,WMAX)).OR.
& (MSAOWH(WLIN,WMAX).LE.MSAIWH(WLIN,WMIN)).OR.
& (MSAOWH(WSAM,WMIN).GE.MSAIWH(WSAM,WMAX)).OR.
& (MSAOWH(WSAM,WMAX).LE.MSAIWH(WSAM,WMIN))) CALL MDWARN(
- 'NO MSS DATA WITHIN WINDOW')

C
C
C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
      IF(NDOTOTL.NE.0) GO TO 900
      IF(NWINDOW.LT.0) CALL OPRPRC      & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
      NWINDOW=1ABS(NWINDOW)
      NPAGE=0

C
C

```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRCHAP
003

C GENERATE WINDOW MAP

C

IF(NSUBW.NE.0) GO TO 400
CALL MAPRNT
GO TO 900

C

C

C GENERATE GEOGRAPHIC SUBWINDOW MAPS

C

400 IF(((KSYOWM(WTIC).NE.'DEG').AND.
& (KSYOWM(WTIC).NE.'MIN')) GO TO 800 & NOT GEOGRAPHIC
CALL SUBWIN(GEDOWM.NSUBW.MAPRNT)
GO TO 900

C

C

C GENERATE UTM SUBWINDOW MAPS

C

800 IF(((KSYOWM(WTIC).NE.'KM ').AND.
& (KSYOWM(WTIC).NE.'MET')) GO TO 800 & NOT UTM
CALL SUBWIN(UTMOWM.NSUBW.MAPRNT)
GO TO 900

C

C

C INVALID TICK INTERVAL FOR GENERATING SUBWINDOW MAPS

C

800 CALL MDWARN('SUBWINDOWS NOT ALLOWED FOR CURRENT PRIMARY TICKS')

C

C

C ANY DIAGNOSTICS???

C

900 IF(INDFATL.EQ.0) GO TO 920
CALL MDNOTE('FATAL ERRORS -- NO MAP GENERATED')
GO TO 990
920 IF(INDWARN.EQ.0) GO TO 990
CALL MDNOTE('PREVIOUS WARNINGS -- NO MAP GENERATED')
IF(MBATCH.EQ.0) WRITE(6,925)
925 FORMAT(4X,'*TRY AGAIN!')
CALL MDCLR(NULCST)
990 WRITE(6,995)
995 FORMAT('0 ')
KOND=' '
RETURN
END

SUBROUTINE PRCXQT 3 INITIALIZATION ROUTINE FOR PRTCLASS

C
C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 05/16/74 ORIGINAL CODE
C E H SCHLOSSER LEC 01/31/79 ALLOW DEFAULT COMMANDS FROM MACDAM
C E H SCHLOSSER LEMSCO 05/16/80 LIMCH(1)=1
C
C
C METHOD
C -----
C
C INITIALIZE PROGRAM. OPEN FILES. IDENTIFY SCENE. QUEUE DEFAULT COMMANDS.
C
C MACHINE-DEPENDENT CODE
C -----
C
C UNIVAC EXEC-8 PROGRAM FILE NAMING CONVENTIONS.
C
C EXTERNAL REFERENCES
C -----
C
C NVIATO 3 NAME NEXT 'VIA' & 'TO' SUBROUTINES
C PSTART 3 PROGRAM START INITIALIZATION
C OPNIZN 3 OPEN INPUT DETECTION FILES (21. 22.)
C CLSHDG 3 PRINT CLASSIFICATION HEADING
C SYSADD 3 ADD DISK SYMBOLIC FILE OR ELT TO SYSIN RUNSTREAM
C MCFATL 3 SUBMIT FATAL DIAGNOSTIC MESSAGE
C
C
C EXCEPTIONS
C -----
C
C 1. MISSING DEFAULT COMMANDS GENERATE A FATAL DIAGNOSTIC.
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST 3 COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
C INCLUDE KOMKLS.LIST 3 COMMON CLASSIFICATION INFO
C EXTERNAL PRCOOD.NULSUB
C
C
C LOCAL DECLARATIONS
C -----
C
C INTEGER LOCFIL 3 LOCATION WITHIN DISK SYMBOLIC FILE (IF > 0)
C
C
C PROCEDURE

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PRCXT
002

```
C -----  
C  
C  
C IDENTIFY PROGRAM  
C     CALL PSTART( 'DAM PRTCLASS(0009)')  
C  
C  
C ON RETURN, CALL PRC000 TO GET DFFAULT/USER COMMANDS  
C     CALL NVIAT0( PRC000.NULSUB)  
C  
C  
C OPEN DETECTION FILE(S) AND IDENTIFY ERTS SCENE  
C     CALL OPN12N  
C     CALL CLSH00( 8)  
C  
C  
C INITIALIZE LIMIT CHANNEL TO FIRST DETECTION CHANNEL  
C     NLINCH=1  
C     LIMCH(1)=1  
C  
C  
C QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE  
C     CALL SYSADD(LOCFIL, 'MACDAM','DEF-PRTCLASS', ' ' )  
C     IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL, 'DAM','DEF-PRTCLASS', ' ' )  
C     IF(LOCFIL.LE.0) CALL MDFATL( 'NO DEFAULT COMMANDS')  
C  
C  
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE  
C  
C     RETURN  
C     END
```

SUBROUTINE HAPRNT 8 MAP DETECTION FILE DATA FOR PRCLASS

```
C
C
C
C HISTORY
C -----
C
C     E H SCHLOSSER      LEC      07/22/73      ORIGINAL CODE
C     E H SCHLOSSER      LEC      03/19/79      'DEN'/'RAD'/'CLA' DETECTION FILES
C     D A BECK           LEC      12/30/79      REVISE FOR PXBDEF FORMAT BUFFERS
C     J C CRISP          LEMSCO   05/18/80      READ CHANNEL LINC1. INSTEAD OF 1
C     J C CRISP          LEMSCO   08/18/80      ADD 4 WORDS TO PRINT BUFFER
C
C
C METHOD
C -----
C
C     GENERATE TICK TABLE IN MEMORY AND TABULAR DATA ON MAP UNIT 0.
C     BREAK THE PRINT WINDOW INTO SECTION(S) MALTH UNIT(S) BY
C     (KPAGE-8) COLUMNS WIDE. PROCESS A SECTION OF WINDOW AT A TIME.
C     WITHIN EACH SECTION. READ/MASK/SCREEN/RESAMPLE DETECTION SCAN LINE(S)
C     TO PRINT LINES AND SYMBOLIZE/TICK/OUTPUT PRINT LINES.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C     ASSUMES 8 CHARS PER INTEGER BIN.
C     ARGUMENTS IN CALLS TO ROUTINES UTILIZING EXEC-8 ER ROUTINES MAY
C     BE MACHINE DEPENDENT.
C
C
C EXTERNAL REFERENCES
C -----
C
C     A4P           8 ADJUSTED FROM PPD COORD
C     P4A           8 PPD COORD FROM ADJUSTED
C     NITHD0        8 PRINT MAP UNIT HEADING
C     PRSYML        8 PRINT SYMBOL LEGEND
C     GENTIC        8 GENERATE TICK TABLE
C     READ2N        8 READ DETECTION FILE(S)
C     MSKPIX        8 MASK PIXELS EXTERIOR TO WINDOW
C     RESPRC        8 RESAMPLE DETECTION PIXELS
C     PROVFI        8 PRINT/OVERPRINT FILE(S)
C     HDFATL        8 PRINT/COUNT/LOG 'FATAL ERROR' MESSAGES
C     HDWARN        8 PRINT/COUNT/LOG 'WARNING' MESSAGES
C     ERPRTA        8 PRINT MESSAGE ON ALTERNATE PRINT FILE
C     DOUBLE PRECISION CBS4CS 8 VARIABLE LENGTH CST FOR FIXED LENGTH CST
C     VIA           TO
C     EXTERNAL RESPRC.  GETINT.GETBYT.GETICE.GETNUL
C
C
C EXCEPTIONS
C -----
C
C     1. IF THE NUMBER OF COLUMNS TO BE PRINTED IS GREATER THAN
```

```

C      THE NUMBER OF COLUMNS PER PAGE TIMES MALTHI THEN
C      GENERATE A WARNING.
C
C      2. IF THE COUNTS PER PIXEL IS < ZERO OR > 20 THEN
C      GENERATE A WARNING.
C
C      3. IF THE ORIGIN TYPE IS NOT 'SCA' OR 'DEG' OR 'MIN' OR
C      'KM' OR 'MET' THEN GENERATE A FATAL ERROR.
C
C      4. IF THE BIN TYPE OR THE DETECTION BUFFER IS NOT 'INT' OR 'BYT'
C      OR 'CHR' THEN GENERATE A FATAL ERROR.
C
C      5. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE MAP.
C
C      6. THE FOLLOWING VALUES OF ISTAT PRODUCE THE FOLLOWING RESULTS
C
C      CONDITION  NEAT LINE  PRINT CELL  DIAGNOSTIC  ACTION
C                CHAR      SYMBOL
C
C      'EOF'      ':'      NO DATA(':')  NONE      PRINT LINE
C      'BADR'     '?'      NO DATA(':')  NONE      PRINT LINE
C      'BADF'     'N/A'    N/A           FATAL     RETURN
C      'OFL'     'N/A'    N/A           FATAL     RETURN
  
```

GLOBAL DECLARATIONS

```

-----
C
C      INCLUDE KONXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C      INCLUDE KONKLS.LIST     & COMMON CLASSIFICATION INFO
C      INCLUDE WINDEF.LIST     & DEFINE STRUCTURE OF WINDOW PACKETS
C      INCLUDE KONDET.LIST     & COMMON DETECTION FILE WINDOWS
C      INCLUDE KONOWM.LIST     & COMMON OUTPUT WINDOW PACKETS
C      INCLUDE KOALTI.LIST     & COMMON ALTERNATE PRINT PARAMETERS
C      INCLUDE KONSYM.LIST     & COMMON SYMBOL TABLE
C      INCLUDE KONTBL.LIST     & COMMON TICK/FREQ TABLE
C      INCLUDE NITAB.LIST     & DEFINE PROCEDURE TO COMPUTE ALT PRT UNIT NUMBERS
C      INCLUDE PXBDEF.LIST     & DEFINE PIXEL BUFFER STRUCTURE
C      INCLUDE PRCDEF.LIST     & DEFINE PRINT CLASS PARAMETERS
  
```

LOCAL DECLARATIONS

```

-----
C
C      INTEGERS IN DET BUF = #INTS PREAMBLE * (NBINS-3)/4 * #EXTRA INTS
C      PARAMETER NWIDBF = 2*(PXBINS-1) * (3548+2-3)/4 * 27
C
C      WDS PRT BUF=WDS PREAM*(MAX FILES*(MAX COLS/PO-NEATLN COLS-MARG COLS))+4
C      PARAMETER NWIPBF=(PXBINS-1)*(MALTHI * (KPAQHI- 2 * 4)) * 4
C
C      INTEGER IDETBF(NWIDBF) & DETECTION FILE BUFFER (PXBDEF FORMAT)
C      INTEGER IPRTBF(NWIPBF) & PRINT FILE BUFFER (PXBDEF FORMAT)
C      INTEGER IPLMIN,IPLMAX & MINIMUM, MAXIMUM PRINT LINES IN WINDOW
C      INTEGER IPCHIN,IPCHAX & MINIMUM, MAXIMUM PRINT COLUMNS IN WINDOW
C      INTEGER IPCLO,IPCMI & LOW, HIGH PRINT COLUMNS IN SECTION
C      INTEGER IPCINC & NUMBER OF PRINT COLUMNS PER SECTION OF WINDOW
  
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

HAPRNT
003

INTEGER NUNIT & UNIT NUMBER OF ALTERNATE PRINT FILE
INTEGER NSALIN & CURRENT MSA LINE NUMBER
INTEGER IPLIN & CURRENT PRINT LINE NUMBER
INTEGER MSASLO,MSASHI & LOW, HIGH MSA SAMPLE NUMBER IN SECTION
INTEGER ISTAT & I/O STATUS RETURN WORD
INTEGER IPLNXS & PRINT LINE NUMBER FOR NEXT SCAN LINE
REAL ADJLIN,ADJSAM & ADJUSTED LINE, SAMPLE
REAL PPDOLN,PPDCOL & PPD LINE, COLUMN
INTEGER NTLCHR & NEAT LINE CHARACTER
INTEGER HROLCS,HORCRS & MARGIN LEFT, RIGHT CHAR STRING

C
C
C DECLARATIONS WHICH ARE GLOBAL TO HAPRNT'S INTERNAL SUBROUTINES
C

PARAMETER NSAVMX=80 & MAXIMUM # OF SAVE LOCATIONS IN ISAVSY
INTEGER ISAVSY(NSAVMX,2) & SAVE TBL OF BIN#/SYMBOLS REPLACED BY TICKS/HALO
INTEGER NSAVED & # OF SYMBOL(S) REPLACED BY TICK(S) & SAVED
INTEGER IPLTIC,IPCTIC,JSY TIC & TICK LINE, COLUMN, AND SYMBOL
INTEGER JPSPEC & PRINT SPEC FOR PROVF!

C
C
C PROCEDURE
C -----
C

CALL TRACE

C
C
C INITIALIZE MINIMUM AND MAXIMUM PRINT LINES AND COLUMNS IN WINDOW
C

IPLMIN=PPDOWN(WLIN,WMIN)
IPLMAX=PPDOWN(WLIN,WMAX)
IPCHIN=PPDOWN(WCOL,WMIN)
IPCMAH=PPDOWN(WCOL,WMAX)

C
C
C BREAK WINDOW INTO UNITS, EACH NOT MORE THAN KPAGE-8 COLUMNS WIDE
C

NITMAX=(1+(IPCMAH-IPCHIN)/(KPAGE-8))
IF(NITMAX.GT.MALTM) CALL MDWARN('WINDOW TOO WIDE')

C
C
C CHECK FOR VALID MAXIMUM COUNT PER PIXEL
C

IF((KTIPIX.LT.0).OR.(KTIPIX.GT.20)) CALL MDWARN('INVALID COUNT PER PIXEL')

C
C
C CHECK FOR DIAGNOSTICS
C

IF(NDOTL.NE.0) GO TO 900

C
C
C CHECK FOR VALID ORIGIN TYPE
C

IF(KSYOWH(WORIO).EQ.'SCA') GO TO 100
IF((KSYOWH(WORIO).EQ.'DEO').OR.(KSYOWH(WORIO).EQ.'MIN')) GO TO 120

```
IF((KSYOH(WORIG).EQ.'KM').OR.(KSYOH(WORIG).EQ.'MET')) GO TO 140  
CALL HOFATL('INVALID ORIGIN TYPE IN HAPRNT')  
GO TO 900
```

C
C
C

PRINT WINDOW NUMBER AND ORIGIN

```
100 WRITE(6,105) NWNDOW,MSAOWH(WLIN,WORIG),MSAOWH(MSAM,WORIG)  
105 FORMAT(6X,'WINDOW #'.13,' (ORIGIN '.14,' LINE. '.14,' SAMPLE)')  
GO TO 150  
120 WRITE(6,125) NWNDOW,GEDOWH(WLAT,WORIG),GEDOWH(WLON,WORIG)  
125 FORMAT(6X,'WINDOW #'.13,' (ORIGIN '.F9.4,' LAT. '.F9.4,' LON)')  
GO TO 150  
140 WRITE(6,145) NWNDOW,UTMOWH(WEA,WORIG),UTMOWH(WNO,WORIG)  
145 FORMAT(6X,'WINDOW #'.13,  
6 ' (ORIGIN '.-3P.F8.3,' KM E. '.F8.3,' KM N)')
```

C
C
C

GENERATE TABULAR DATA (UNIT 0)

```
150 NUNIT=0  
IF(NLEGD.EQ.0) GO TO 200      & NO UNIT 0 UNLESS LEGEND MODE SWITCH IS ON  
NITLO=0  
NITHI=0  
INCLUDE NITROT.LIST  
NIT=0  
NUNIT=NTAB(NIT)  
CALL NITDO( NUNIT)      & PRINT UNIT HEADING  
CALL PRSYML( NUNIT)      & PRINT SYMBOL LEGEND  
IF((KTIPIX.GE.1).AND.(KTIPIX.LE.9))  
6 WRITE(NUNIT,175) KTIPIX  
175 FORMAT('011 COUNT = 1/''.11,' PIXEL)')  
IF(KTIPIX.GE.10)  
6 WRITE(NUNIT,185) KTIPIX  
185 FORMAT('011 COUNT = 1/''.12,' PIXEL)')  
200 CALL GENTIC( NUNIT)      & GENERATE TICK TABLE AND PRINT ON NUNIT IF NOT 0
```

C
C
C

INITIALIZE LOW/HIGH PRINT COLUMNS FOR FIRST SECTION OF WINDOW

```
IPCINC=(KPAGE-6)*MALTM  
IPCLO=IPCHIN  
IPCHI=MIND((IPCLO-IPCINC-1),IPCHAX)
```

C
C
C

PROCESS WINDOW IN SECTIONS. EACH NOT MORE THAN MALTM PRINT UNITS WIDE

```
DO 600 NITLO=1,NITHAX,MALTM      & LOW UNIT FOR EACH SECTION OF WINDOW  
NITHI=MIND((NITLO-MALTM-1),NITHAX)      & HIGH UNIT  
INCLUDE NITROT.LIST & ROTATE THE ASSIGN OF LOGICAL UNITS TO MAP UNITS
```

C
C
C

INITIALIZE FIRST TICK. EXACT FIRST PPD LINE#. APPROXIMATE FIRST SCAN LINE#

```
CALL INITIC(IPLTIC,IPCTIC,JSYTIC)  
IPLIN=IPLMIN
```



```

      CALL A4P(ADJLIN,ADJSAM,      8 INSURE THAT MSALIN
      FLOAT(IPLMIN-1),0.)      8 <= FIRST SCAN LINE
      MSALIN=ADJLIN              8 WHICH MAPS TO IPLMIN
C
C
C PRINT UNIT HEADINO(S)
C
      DO 300 NIT=NITLO,NITHI
      NUNIT=NTAB(NIT)      8 COMPUTE LOGICAL UNIT FOR MAP UNIT
      CALL NITHOO( NUNIT)
300   CONTINUE
C
C
C GENERATE AND PRINT TOP COLUMN SCALE/BORDER FOR UNIT(S) IN CURRENT SECTION
C
      CALL SAMSCL(IPTBF,(IPLMIN),(IPCLO),(IPCMI))
      NUNIT=NTAB(NITLO)
      IPTBF(PXREC)=NWINDOW*1000
      CALL PROVF(NUNIT,      '.4.' '.4.' '.011111'.IPTBF)
C
C
C PROCESS ALL PRINT LINES IN CURRENT SECTION OF WINDOW
C
400   CONTINUE      8 DO 540 UNTIL IPLIN > IPLMAX
C
C
C COMPUTE LOW AND HIGH DETECTION SAMPLES FOR CURRENT PRINT LINE
C
      CALL A4P(ADJLIN,ADJSAM,      FLOAT(IPLIN),FLOAT(IPCLO))
      MSASLO=ADJSAM      8 MUST BE <= MSA SAMPLE WHICH MAPS TO IPCLO
      CALL A4P(ADJLIN,ADJSAM,      FLOAT(IPLIN),FLOAT(IPCMI+1))
      MSASHI=ADJSAM+1.0      8 MUST BE >= MSA SAMPLE WHICH MAPS TO IPCMI
      CALL PRINI(IPTBF,(IPLIN),(IPCLO),(IPCMI))
C
C
C READ/MASK/SCREEN/RESAMPLE ANY SCAN LINE(S) WHICH MAP TO CURRENT PRINT LINE
C
410   CONTINUE      8 DO 400 WHILE IPLNXS <= IPLIN
      CALL P4A(PPDLIN,PPCOL,FLOAT(MSALIN),0.)
      IPLNXS=PPDLIN
      IF(IPLNXS.GT.IPLIN) GO TO 500      8 NO MORE MSA LINES FOR IPLIN
      IF(IPLNXS.LT.IPLIN) GO TO 400      8 NOT YET MSA LINES FOR IPLIN
      8 GET MSA LINE FOR IPLIN
C --READ
      CALL READZ(IDE7BF,(NWINDOW),ISTAT,
      MSALIN,LINCHI,MSASLO,MSASHI)
      IF(ISTAT.NE.'BADF').AND.
      (ISTAT.NE.'OFL') GO TO 430
      CALL M0FATL(CBS4CS(ISTAT,1,4),
      'WHILE READING DETECTION FILE')
      CALL ERPTA('10      '.2.6.
      '*I/O ERROR IGNORE OUTPUT*')
      GO TO 500
430   NTLCHR=':
      IF(ISTAT.EQ.'BADR') NTLCHR='?'
C --MASK
      CALL MSKPIX(IDE7BF, IDE7BF)

```

```

C --SCREEN & RESAMPLE
      IF(I0ETBF(PXBINT).NE.'INT') GO TO 440
      CALL RESPRC(IPRTBF,
        IPRTBF,I0ETBF,GETINT)
      GO TO 480
440    IF(I0ETBF(PXBINT).NE.'BYT') GO TO 450
      CALL RESPRC(IPRTBF,
        IPRTBF,I0ETBF,GETBYT)
      GO TO 480
450    IF(I0ETBF(PXBINT).NE.'CHR') GO TO 460
      CALL RESPRC(IPRTBF,
        IPRTBF,I0ETBF,GETICE)
      GO TO 480
460    IF(I0ETBF(PXBINT).NE.'NUL') GO TO 470
      CALL RESPRC(IPRTBF,
        IPRTBF,I0ETBF,GETNUL)
      GO TO 480
470    CALL HDPATL(C0S4CS(I0ETBF(PXBINT),1,3),
      ' IS INVALID BIN TYPE FOR I0ETBF IN HAPRNT')
      GO TO 800
C --LOOP TO GET NEXT SCAN LINE FOR CURRENT PRINT LINE
480    MSAL:N=MSALIN+1
      GO TO 410

C
C
C SYMBOLIZE PPD CELLS IN CURRENT PRINT BUFFER
C
500    CALL SYMPC(IPRTBF)
C
C
C OUTPUT PRINT LINES FROM CURRENT SYMBOLIZED PRINT BUFFER UNTIL SCAN LINE
C FOR NEW PRINT LINE IS AVAILABLE
C
520    CONTINUE      & DO UNTIL IPLIN >= IPLNXL
      CALL TICK(IPRTBF)      & INSERT TICKS FOR IPLIN
      CALL CSTYIN(MRGLCS,(1),4, IPLIN,4,'0')
      MRGRCS=MRGLCS
      NUNIT=NTAB(INITLO)
      IPRTBF(PXRECNI)=IPRTBF(PXRECNI)+1
      CALL PROVF(NUNIT,
        MRGLCS,4,MRGRCS,4,NTLCHR,JPSPEC,IPRTBF)
      CALL FIXSYM(IPRTBF)      & REMOVE TICKS FOR IPLIN
      IPLIN=IPLIN+1
      IF(IPLIN.GE.IPLNXL) GO TO 540
      GO TO 520      & LOOP TO OUTPUT ANOTHER PRINT LINE

C
C
C LOOP TO GET NEXT SCAN LINE FOR NEW PRINT LINE
C
540    IF(IPLIN.GT.IPLMAX) GO TO 800
      GO TO 480

C
C
C GENERATE AND PRINT BOTTOM SCALE/BORDER FOR UNIT(S) IN CURRENT SECTION
C
600    CALL SAMSCL(IPRTBF,(IPLMAX),(IPCLO),(PCMI))
  
```



```
C
C   INITIC: INITIALIZE TICK POINTER AND OBTAIN FIRST TICK.
C   GETIC: OBTAIN NEXT TICK.
C
C MACHINE-DEPENDENT CODE
C
C   NONE.
C
C EXTERNAL REFERENCES
C
C   NONE.
C
C EXCEPTIONS
C
C   1. INITIC MUST BE CALLED BEFORE ANY CALL TO GETIC.
C
C GLOBAL DECLARATIONS
C
C   INCLUDE KOMTBL.LIST      & COMMON TICK TABLE AND FUNCTIONS
C
C LOCAL DECLARATIONS
C
C   INTEGER JSYD1(2)/'*', '**/'      & TICK SYMBOLS FOR TICK LEVELS 0 & 1
C   INTEGER NTICK      & TICK POINTER
C
C PROCEDURE
C
C INITIALIZE TICK POINTER. THEN GET FIRST TICK
C
C   NTICK=0
C
C
C
C   ENTRY GETIC:      & GET NEXT TICK
C   0 IPLTIC.        & INTEGER PRINT LINE FOR TICK
C   0 IPCTIC.        & INTEGER PRINT COLUMN FOR TICK
C   0 JSYTIC)        & TICK SYMBOL:  '*'-PRIMARY,  '**'-SECONDARY
C
C INCREMENT TICK POINTER
C
C   NTICK=NTICK+1
C
C GET TICK LINE, COLUMN, AND LEVEL
C
C   IPLTIC=LINTIC(NTICK)
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

HAPRNT
011

```
C
C HISTORY
C
C     E M SCHLOSSER   LEC   07/22/73   ORIGINAL CODE
C     E M SCHLOSSER   LEC   03/19/79   'DEN'/'RAD'/'CLA' DETECTION FILES
C     D A BECK        LEC   12/30/79   REVISE FOR PXBDEF FORMAT BUFFERS
C
C METHOD
C     LOOK UP SYMBOL AND SYMBOLIZE THE PRINT BUFFER.
C
C EXTERNAL REFERENCES
C     NONE.
C
C EXCEPTIONS
C     NONE.
C
C GLOBAL DECLARATIONS
C     INCLUDE PXBDEF.LIST      & DEFINE PIXEL BUFFER STRUCTURE
C     INCLUDE KOSY.M.LIST     & COMMON SYMBOL TABLE
C
C LOCAL DECLARATIONS
C     INTEGER IPRTBF(1)      & ARGUMENT
C     INTEGER IPBIN, IPLBIN, IPHBIN  & BIN #, LOW, AND HIGH
C
C PROCEDURE
C
C SET BIN POINTERS
C
C     IPLBIN=IPRTBF(PXLBIN)
C     IPHBIN=IPRTBF(PXHBIN)
C
C LOOK UP SYMBOLS FOR SYMBOLIZATION OF BUFFER
C
C     DO 150 IPBIN=IPLBIN,IPHBIN,1
C         IPIXEL=MIND(IPRTBF(PXBINS-1+IPBIN),(KSYMSZ-1))
C         IPRTBF(PXBINS-1+IPBIN)=KSYM(IPIXEL+1)
C     150 CONTINUE
C
C SET PRINT SPEC TO ALLOW OVERPRINTING
C
C     JPSPEC='1000*'
C
```


DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

HAPRNT
014

```
C      E H SCHLOSSER   LEC      03/19/79   'DEN''RAD''CLA' DETECTION FILES
C      D A BECK       LEC      12/30/79   REVISE FOR PXBDEF FORMAT BUFFERS
C
C      METHOD
C
C      INSERT TICK(S) FOR A LINE OF PRINT AND SAVE SYMBOL(S)
C      REPLACED BY TICK(S).
C
C      MACHINE-DEPENDENT CODE
C
C      NONE.
C
C      EXTERNAL REFERENCES
C
C      INTEGER ICE          % INTEGER-CHARACTER-EQUIVALENT OF 1ST CHAR IN STRING
C
C      EXCEPTIONS
C
C      1. INITIC MUST BE CALLED BEFORE FIRST CALL TO TICK ROUTINE.
C
C      GLOBAL DECLARATIONS
C
C      INCLUDE PXBDEF.LIST      % DEFINE PIXEL BUFFER STRUCTURE
C
C      LOCAL DECLARATIONS
C
C      INTEGER IPRTBF(1)      % ARGUMENT
C      INTEGER IPBTIC        % BIN # OF TICK
C      INTEGER ICESPA        % I-C-E OF ' '
C      INTEGER NBINCO        % BIN # OF COLUMN ZERO
C
C
C      PROCEDURE
C
C      INITIALIZE POINTER FOR SAVE TABLE OF SYMBOLS REPLACED BY TICKS
C
C      NBINCO=IPRTBF(PXLBIN)-IPRTBF(PXLCOL)  % FOR INSERTING TICKS
C      NSAVED=0
C      ICESPA=ICE(' ')
C
C
C      GET AND INSERT TICKS
C
C      500 IPBTIC=IPCTIC+NBINCO
C          IF (IPLTIC.GT.IPLIN) GO TO 900      % SAVE TICK FOR SUBSEQUENT LINE
C          IF (
C      % (IPLTIC.LT.IPLIN).OR.                % TICK ABOVE PRINT LINE
C      % (IPBTIC.LT.IPRTBF(PXLBIN)).OR.      % TICK LEFT OF LOW BIN
C      % (IPBTIC.GT.IPRTBF(PXHGIN)))        % TICK RIGHT OF HIGH BIN
```

```

      8 GO TO 550                8 THEN IGNORE IT
      IF (JSYTC.EQ.'*') GO TO 510 8 ALWAYS INSERT PRIMARY TICK
      IF ((ICE(IPRTBF(PXBINS-1+IPBT'C)).EQ.ICSPA) GO TO 510 8 PUT SEC TICK
      IF ((IPRTBF(PXBINS-1+IPBTIC).NE.'*') GO TO 550 8 PUT SEC TICK IN NODAT
510 CONTINUE 8 INSERT LEFT HALO IN 'NO DATA' PPD CELL
      IF ((IPRTBF(PXBINS-1+IPBTIC-1).NE.'*') GO TO 520
      NSAVED=MIND(NSAVED+1,NSAVMX)
      ISAVSY(NSAVED,1)=IPBTIC-1
      ISAVSY(NSAVED,2)=IPRTBF(PXBINS-1+IPBTIC-1)
      IPRTBF(PXBINS-1+IPBTIC-1)=' '
520 CONTINUE 8 INSERT TICK IN PPD CELL
      NSAVED=MIND(NSAVED+1,NSAVMX)
      ISAVSY(NSAVED,1)=IPBTIC
      ISAVSY(NSAVED,2)=IPRTBF(PXBINS-1+IPBTIC)
      IPRTBF(PXBINS-1+IPBTIC)=JSYTC
530 CONTINUE 8 INSERT RIGHT HALO IN 'NO DATA' PPD CELL
      IF ((IPRTBF(PXBINS-1+IPBTIC+1).NE.'*') GO TO 550
      NSAVED=MIND(NSAVED+1,NSAVMX)
      ISAVSY(NSAVED,1)=IPBTIC+1
      ISAVSY(NSAVED,2)=IPRTBF(PXBINS-1+IPBTIC+1)
      IPRTBF(PXBINS-1+IPBTIC+1)=' ' 8 RIGHT HALO
550 CALL GETIC (IPLTIC,IPCTIC,JSYTC)
      GO TO 500
  
```

```

C
C
C RETURN TO CALLING ROUTINE
C
C 900 RETURN
  
```

```

C
C
C INTERNAL
C SUBROUTINE FIXSYM( 8 UNTICK AND FIX SYMBOLS IN PPD BUFFER
C U (IPRTBF) 8 PRINT BUFFER
  
```

C HISTORY

C	E M SCHLOSSER	LEC	07/22/73	ORIGINAL CODE
C	E M SCHLOSSER	LEC	03/19/79	'DEN'/'RAD'/'CLA' DETECTION FILES
C	D A BECK	LEC	12/30/79	REVISE FOR PXBDEF FORMAT BUFFERS

C METHOD

C THIS ROUTINE IS CALLED WHEN (DUE TO RESAMPLING) NO SCAN LINE IS
 C AVAILABLE TO GENERATE THE NEXT PRINT LINE. FIXSYM TAKES THE
 C PPD BUFFER JUST OUTPUT FOR THE CURRENT PRINT LINE. REMOVES THE
 C TICKS, AND 'FIXES IT UP' SO IT CAN BE USED FOR THE NEXT PRINT
 C LINE.

C REPLACE TICK(S) WITH ORIGINAL SYMBOL(S).
 C IF COUNTING, SET NON-BLANK CELLS TO ':' (NO DATA) AND SET PRINT

```

C     SPEC TO SUPPRESS OVERPRINTING.
C
C     MACHINE-DEPENDENT CODE
C
C     ASSUMES 8 CHARS PER INTEGER BIN.
C
C     EXTERNAL REFERENCES
C
C     INTEGER ICE      8 INTEGER-CHARACTER-EQUIVALENT OF 1ST CHAR IN STRING
C
C     EXCEPTIONS
C
C     NONE.
C
C     GLOBAL DECLARATIONS
C
C     INCLUDE PXBDEF.LIST      8 DEFINE PIXEL BUFFER STRUCTURE
C     INCLUDE KONKLS.LIST     8 COMMON CLASSIFICATION INFO
C
C     LOCAL DECLARATIONS
C
C     INTEGER IPRTB(1)      8 ARGUMENT
C     INTEGER ICESPA        8 I-C-E OF ' '
C     INTEGER IPBIN, IPLBIN, IPMBIN      8 BIN #, LOW, AND HIGH
C
C
C     PROCEDURE
C
C     REPLACE TICK(S) & TICK HALO(S), IF ANY, WITH ORIGINAL SYMBOL(S)
C
C     200 IF(NSAVED.LE.0) GO TO 500
C         IPBIN=ISAVSY(NSAVED,1)
C         IPRTB(PXBINS-1+IPBIN)=ISAVSY(NSAVED,2)
C         NSAVED=NSAVED-1
C     GO TO 200
C
C
C     IF COUNTING, SET NON-BLANK PPD CELLS TO ':' (NO DATA) & SUPPRESS OVERPRINTING
C
C     500 IF(KTPIX.EQ.0) GO TO 900
C         IPLBIN=IPRTBF(PXLBIN)
C         IPMBIN=IPRTBF(PXMBIN)
C         ICESPA=ICE(' ')
C         DO 700 IPBIN=IPLBIN,IPMBIN
C             IF(ICE(IPRTBF(PXBINS-1+IPBIN)).NE.ICESPA)
C                 IPRTB(PXBINS-1+IPBIN)=':'
C
C     700 CONTINUE
C     JPSPEC='1.....'      8 NO NEED TO EVEN ATTEMPT OVERPRINTING
C
C
  
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

HAPRNT
017

C RETURN TO CALLING ROUTINE

C

900 RETURN
END

SUBROUTINE NITHDO: 8 PRINT MAP UNIT HEADING
(NUNIT) 8 OUTPUT PRINT UNIT

C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 07/20/73 ORIGINAL CODE
C E H SCHLOSSER LEMSCO 01/20/80 CALL PRNUM INSTEAD OF PRNUM
C E H SCHLOSSER LEMSCO 05/16/80 NEGATIVE MAP NUMBER 8 USE OETICE
C
C
C METHOD
C -----
C
C WRITE BOX HEADING, SCENE IDENTIFICATION, CLASSIFICATION SUMMARY,
C MAP WINDOW IDENTIFICATION, AND CONTROL SUMMARY ON NUNIT.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C NONE.
C
C
C EXTERNAL REFERENCES
C -----
C
C NDUNIT 8 PAGE EJECT 8 PRINT PAGE HEADING LINES ON SPECIFIED UNIT
C OETICE 8 GET INTEGER-CHARACTER-EQUIVALENT FROM CHAR IN STRING
C PRNUM 8 PRINT BOX NUMBERS ON SPECIFIED UNIT
C CLSHDO 8 PRINT CLASSIFICATION HEADING ON SPECIFIED UNIT
C MAPHDO 8 PRINT MAP WINDOW HEADING INFORMATION ON SPECIFIED UNIT
C INTEGER ICE 8 INTEGER-CHARACTER-EQUIVALENT FROM CHARACTER
C
C
C EXCEPTIONS
C -----
C
C NONE.
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C INCLUDE KOMALT.LIST 8 COMMON ALTERNATE PRINT FILE POINTERS
C
C
C LOCAL DECLARATIONS
C -----
C
C INTEGER IMH 8 HOURS UNITS DIOIT
C INTEGER IMID 8 MINUTES TENS DIOIT
C INTEGER IMI 8 MINUTES UNITS DIOIT

INTEGER NM100 & WINDOW NUMBER HUNDREDS DIGIT
 INTEGER NM10 & WINDOW NUMBER TENS DIGIT
 INTEGER NM1 & WINDOW NUMBER UNITS DIGIT
 INTEGER ICE0 & INTEGER-CHARACTER-EQUIVALENT OF '0'
 PARAMETER BLANK=00 & BOX NUMBER PRINTS AS BLANK

```

C
C
C PROCEDURE
C -----
C
C
C     CALL TRACE
C
C
C BEGIN HEADING
C
C     NPAGE=0
C     CALL MDUNIT(4,NUNIT)
C
C
C SET UP HOUR AND MINUTE BOX NUMBERS
C
C     ICE0=ICE('0')
C     CALL GETICE(1MI, JHMS,2)
C     MI=MI-ICE0
C     CALL GETICE(1M10, JHMS,3)
C     M10=M10-ICE0
C     CALL GETICE(1MI, JHMS,4)
C     MI=MI-ICE0
C
C
C SET UP MAP NUMBER BOX NUMBERS
C
C     NM100=NMWINDOW/100        & 100'S DIGIT
C     NM10=(NMWINDOW-100*NM100)/10        & 10'S DIGIT
C     NM1=(NMWINDOW-100*NM100)-10*NM10        & 1'S DIGIT
C     NM100 = NM100 + 10        & NEGATIVE (WHITE ON BLACK)
C     NM10 = NM10 + 10
C     NM1 = NM1 + 10
C
C
C PUT BOX NUMBERS ON WIDE PAGE
C
C     IF(KPAGE.LT.120) GO TO 300
C     CALL PRNUM(NUNIT,16,12,1MI,1MI,NM100,NM10,NM1,NIT,BLANK)
C     WRITE(NUNIT,135)
C     FORMAT(1X/
C 1        &X,0(' '),10X,0(' '),10X,4(' '),10X,0(' ') /
C 2        10X,'HOUR',13X,'MINUTE',20X,'MAP NUMBER',29X,'UNIT')
C     GO TO 700
C
C
C PUT BOX NUMBERS ON NARROW PAGE
C
C 300 IF(KPAGE.LT.80) GO TO 500
C     CALL PRNUM(NUNIT,16,12,NM100,NM10,NM1,NIT,BLANK,BLANK,BLANK)
  
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

NITHOO
003

```
      WRITE(NUNIT,339)
339   FORMAT(1X/
        1   6X,44(' '),10X,8(' ')//
        2   29X,'MAP NUMBER',29X,'UNIT')
      GO TO 700

C
C
C PUT BOX NUMBERS ON VERY NARROW PAGE
C
500 CALL PRNUM(NUNIT,15,12,NW16,NW1,NIT,BLANK,BLANK,BLANK,BLANK)
      WRITE(NUNIT,539)
539   FORMAT(1X/
        1   6X,26(' '),10X,8(' ')//
        2   16X,'MAP NUMBER',26X,'UNIT')

C
C
C COMPLETE HEADING
C
700 CALL CLSHOO(NUNIT)
      CALL MAPHOO(NUNIT)
      NLINE=99
      RETURN
      END
```


SUBROUTINE OPRPRC 3 OPEN ALTERNATE PRINT FILES FOR PRCLAS

C
C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 08/28/78 ORIGINAL CODE
C J C CRISP LEC 12/18/79 REVISE TO INCLUDE PRCLAS-PROC
C
C
C METHOD
C -----
C
C A MAXIMUM OF 9 ALTERNATE PRINT FILES ARE OPENED & INITIALIZED.
C
C
C MACHINE-DEPENDENT CODE
C -----
C
C NONE.
C
C
C EXTERNAL REFERENCES
C -----
C
C OPENPR OPEN ALTERNATE PRINT FILES
C
C
C EXCEPTIONS
C -----
C
C 1. ONE PRINT FILE IS ALWAYS OPENED. UNLESS IN DATA/CHECKOUT MODE.
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST 3 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C INCLUDE PRCLAS.LIST 3 DEFINE MALTHI AND KPAGHI FOR PRCLAS
C
C
C PROCEDURE
C -----
C
C CALL TRACE
C
C
C NO MORE THAN MALTHI ALTERNATE PRINT FILES FOR PRCLAS
C
C MALTH=MAX0(MALTH,1)
C MALTH=MIN0(MALTH,MALTHI)
C
C
C NO MORE THAN KPAGHI COLUMNS PER PAGE IN PRCLAS
C

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

OPRPRC
002

```
      KPAGE=MAX0(KPAGE,1)
      KPAGE=MIN0(KPAGE,KPAOH)
C
C
C OPEN FILE(S)
C
      IF(MDATA0.EQ.0) CALL OPENPR
C
C
      RETURN
      END
```


C A FATAL ERROR WILL BE ISSUED.

C
C
C
C
C
C

GLOBAL DECLARATIONS

C
C INCLUDE KOHNER.LIST : COMMON ERTS SCENE PARAMETERS
C INCLUDE KOMFIT.LIST : COMMON ADJUSTMENT/REGISTRATION PARAMETERS
C INCLUDE KOMKLS.LIST : COMMON CLASSIFICATION INFO
C INCLUDE TRFORM.LIST : DEFINE COORDINATE TRANSFORMATION FUNCTIONS
C INCLUDE PXRDEF.LIST : DEFINE PIXEL BUFFER STRUCTURE

C
C
C
C
C

LOCAL DECLARATIONS

C
C INTEGER IPRTBF(1) : ARGUMENT
C INTEGER IDETBF(1) : ARGUMENT
C INTEGER MSASAM : SAMPLE NUMBER OF PIXEL BEING SCREENED/RESAMPLED
C INTEGER MSALIN : MSS LINE NUMBER
C INTEGER MSASLO : MSS LOW SAMPLE
C INTEGER MSASHI : MSS HIGH SAMPLE
C INTEGER NODADE : DETECTION BUFFER NO DATA THRESHOLD
C INTEGER NODAPR : PRINT BUFFER NO DATA THRESHOLD
C INTEGER NRDC : RADIANCE/DENSITY/CLASS BEING SCREENED/RESAMPLED
C REAL CORLIN : CORRECTED LINE NUMBER FOR MSALIN
C REAL CORSAM : CORRECTED SAMPLE NUMBER FOR MSASAM
C REAL ADJLIN : ADJUSTED LINE NUMBER FOR MSALIN
C REAL ADJSAM : ADJUSTED SAMPLE NUMBER FOR MSASAM
C INTEGER IPCLO,IPCMI : LOW AND HIGH PRINT COLUMNS
C INTEGER IPLIN : PRINT LINE
C INTEGER IPCOL : PPD COLUMN FOR MSASAM
C INTEGER IPC1,IPC2 : PRINT COLUMNS FOR FIRST AND LAST NODATA PIXELS
C INTEGER NBINSO : BIN CONTAINING SAMPLE 0 IN IDETBF
C INTEGER NBINCO : BIN CONTAINING COLUMN 0 IN IPRTBF
C INTEGER IPBIN : BIN POINTER IN IPRTBF

C
C
C
C
C

PROCEDURE

C
C
C
C
C
C
C

C CHECK THAT SAME ACTUAL ARG WAS USED FOR IPRTBF AND IPBUF

C IF(LOC(IPRTBF).NE.LOC(IPBUF)) CALL MDFATL(
C = 'IPRTBF & IPBUF NOT SAME IN RESPRC')

C
C
C
C

C COMPUTE BIN NUMBER FOR SAMPLE AND COLUMN 0 AND BIN POINTER

C NBINSO=IDETBF(PXLBIN)-IDETBF(PXLSAM)
C NBINCO=IPRTBF(PXLBIN)-IPRTBF(PXLCOL)
C IPBIN=PXBIN-1+NBINCO

C

```

C
C COMPUTE SAMPLES FOR LOW AND HIGH PRINT COLUMNS
C
  IPLIN=IPRTBF(PXLINO)
  IPCLO=IPRTBF(PXLCOL)
  IPCHI=IPRTBF(PXHCOL)
  CALL A4PIADJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCLO)
  MSASLO=ADJSAM * MUST BE <= MSA SAMPLE WHICH MAPS TO IPCLO
  CALL A4PIADJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCHI+1))
  MSASHI=ADJSAM+1.0 * MUST BE >= MSA SAMPLE WHICH MAPS TO IPCHI

C
C INITIALIZE NO DATA THRESHOLDS, SCAN AND CORRECTED LINE NUMBER
C
  NODADE=IDETBF(PXNODA)
  NODAPR=IPRTBF(PXNODA)
  MSALIN=IDETBF(PXLINO)
  CORLIN=CORL4A(MSALIN,MSASAM)

C
C INITIALIZE SAMPLE POINTER
C
  MSASAM=MSASLO-1

C
C INITIAL STATE (DETERMINE IF FIRST SAMPLE IS DEFINED)
C
  100 MSASAM=MSASAM+1
  CORSAM=CORS4A(MSALIN,MSASAM)
  IPCOL=PPDC4C(CORLIN,CORSAM)
  IF (IPCOL.LT.IPCLO) GO TO 100
  IF (MSASAM.GT.IDETBF(PXMSAM)) GO TO 700
  IF (MSASAM.GE.IDETBF(PXLSAM)) GO TO 150

C
C INITIAL UNDEFINED STATE (RESAMPLE/STORE INITIAL UNDEFINED PIXELS)
C
  IPC1=IPCOL
  MSASAM=IDETBF(PXLSAM)
  CORSAM=CORS4A(MSALIN,MSASAM-1)
  IPC2=PPDC4C(CORLIN,CORSAM)
  DO 120 IPCOL=IPC1,IPC2
    IF (IPCOL.GT.IPCHI) GO TO 900
    IPRTBF(IPBIN+IPCOL)=NODAPR
  120 CONTINUE
  CORSAM=CORS4A(MSALIN,MSASAM)
  IPCOL=PPDC4C(CORLIN,CORSAM)

C
C INITIAL DEFINED STATE (DETERMINE STATE OF FIRST DEFINED SAMPLE)
C
  150 CALL GETBIN (NRDC, IDETBF(PXBINS),(MSASAM+NBINS0))
  IF(NRDC.GE.NODADE) GO TO 500

C
C BEGIN DATA STATE (DETERMINE WHETHER TO STORE OR COUNT)
C

```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

RESPRC
004

```
      200 IF(KTIPIX.NE.0) GO TO 400      & COUNTING?
C
C DATA STORE SUB-STATE (SCREEN/RESAMPLE/STORE DATA PIXELS)
C
      300 IF((NRDC.LT.LCVLO(1)).OR.(NRDC.GT.LCVHI(1))) GO TO 320
          IPRTBF(IPBIN+PCOL)=NRDC      & IN RAD/DEN/CLA LIMITS
      320 MSASAM=MSASAM+1
          CORSAM=CORS4A(MSALIN,MSASAM)
          IPCOL=PPDC4C(CORLIN,CORSAM)
          IF(MSASAM.GT.MSASHI) GO TO 900
          IF (IPCOL.GT.IPCHI) GO TO 900
          IF (MSASAM.GT.IDETBF(PXMSAM)) GO TO 700
              CALL GETBIN (NRDC, IDETBF(PXBINS),(MSASAM+NBINS0))
              IF(NRDC.LT.NODADE) GO TO 300
          GO TO 500
C
C DATA COUNT SUB-STATE (SCREEN/RESAMPLE/COUNT DATA PIXELS)
C
      400 IF((NRDC.LT.LCVLO(1)).OR.(NRDC.GT.LCVHI(1))) GO TO 420
          IPRTBF(IPBIN+PCOL)=
          & IPRTBF(IPBIN+PCOL)+MIN0(NRDC.KTIPIX)      & IN RAD/DEN/CLA LIMITS
      420 MSASAM=MSASAM+1
          CORSAM=CORS4A(MSALIN,MSASAM)
          IPCOL=PPDC4C(CORLIN,CORSAM)
          IF (MSASAM.GT.IDETBF(PXMSAM)) GO TO 700
          IF(MSASAM.GT.MSASHI) GO TO 900
          IF (IPCOL.GT.IPCHI) GO TO 900
              CALL GETBIN (NRDC, IDETBF(PXBINS),(MSASAM+NBINS0))
              IF(NRDC.LT.NODADE) GO TO 400
C
C NODATA STATE (RESAMPLE/STORE DEFINED NODATA PIXELS)
C
      500 IPRTBF(IPBIN+PCOL)=NODAPR
          MSASAM=MSASAM+1
          CORSAM=CORS4A(MSALIN,MSASAM)
          IPCOL=PPDC4C(CORLIN,CORSAM)
          IF (MSASAM.GT.IDETBF(PXMSAM)) GO TO 700
          IF(MSASAM.GT.MSASHI) GO TO 900
          IF (IPCOL.GT.IPCHI) GO TO 900
              CALL GETBIN (NRDC, IDETBF(PXBINS),(MSASAM+NBINS0))
              IF(NRDC.GE.NODADE) GO TO 500
          GO TO 200
C
C FINAL UNDEFINED STATE (RESAMPLE/STORE FINAL UNDEFINED PIXELS)
C
      700 IPC1=PCOL
          CORSAM=CORS4A(MSALIN,MSASHI)
          IPC2=PPDC4C(CORLIN,CORSAM)
          DO 750 IPCOL=IPC1,IPC2
              IF (IPCOL.GT.IPCHI) GO TO 900
          IPRTBF(IPBIN+PCOL)=NODAPR
      750 CONTINUE
```

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

RESPRC
005

C
C
C FINAL STATE (ALL DONE)
C
900 RETURN
END

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PLTCLASS
001

```
C      PROGRAM PLTCLASS
C      -----
C
C (R E HARVESON)
C
C THIS PROGRAM PRODUCES PLOTS OF CLASSIFIED ERTS MSS DATA ON A PEN
C PLOTTER.  THE USER SPECIFIES THE SCALE, WINDOW DIMENSIONS, LOCATION, ETC.
C OF EACH MAP.
C
C THIS PROGRAM IS LIMITED TO THAT PART OF THE WORLD COVERED BY THE
C CLARKE 1866 SPHEROID (NORTH AMERICA).
C
C THE PROGRAM CLASSIFY MUST BE EXECUTED BEFORE THIS PROGRAM.  PREFERABLY
C IN THE SAME RUN.
C
C
C      INCLUDE KOMXQT.LIST
C      INCLUDE KOMLOG
C      INCLUDE KOMLUS
C      INCLUDE KOML2N
C      INCLUDE KOMNER
C      INCLUDE KOMKLS
C      INCLUDE KOMFIT
C      INCLUDE KOMDET
C      INCLUDE KOMALT
C      INCLUDE KOMSYM
C      INCLUDE KOMPLT
C      INCLUDE KOMIHW
C      INCLUDE KOMOHW
C
C
C      EXTERNAL PLC000.PLCXQT
C
C      CALL NVIATO( PLC000.PLCXQT)      B FIRST CALL IS VIA PLC000 TO PLCXQT
100 CALL VIATO
GO TO 100
END
```


PROGRAM PLTCLASS/VIRTUAL

HISTORY

E H SCHLOSSER	LEC	08/02/74	ORIGINAL CODE
E H SCHLOSSER	LEC	11/06/79	MAP.FZ(N): NO 'N' IN DEMAND

METHOD

CONSTRUCT MAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT BXQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE MAP & BXQT COMMANDS TO TEMPORARY FILE 20.
ADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 6-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS & FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IOWS & INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS & TERMINATE PROGRAM EXECUTION
DAM.PLTCLASS-MAP & SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAPOPT & STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS BASO-D & BPREP-D.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

(2 = REAL TIME
(3 = LOW EXEC
(4 = DEMAND
(5 = DEADLINE BATCH
(6 = BATCH

(BXQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER A5 IN

PLTCLASS OVERLAY STRUCTURE

HISTORY

R E NARVESON LEC 11/07/78 MODIFIED FROM PRTCLASS-MAP
E H SCHLOSSER LEC 02/28/79 MACRO COMMANDS & TIME COMMAND

LIB DAM.

SEG S-MAIN

IN DAM.PLTCLASS/ . MAIN PROGRAM
IN DAM.NVIATO . NAME/CALL 'VIA' AND 'TO' SUBROUTINES
IN DAM.NULSUB . DO NOTHING
IN DAM.SYS-BLOCK . BLOCK DATA SUBROUTINE

. MONITOR FOR PHASE 0.1.2.9 COMMANDS -----

IN DAM.PLC000 . CALL USER-SPECIFIED PHASE 0 ROUTINE
IN DAM.PLC129 . CALL PREVIOUSLY NAMED PHASE 1/2/9 'TO' ROUTINE
IN DAM.NTABS/DAM . DAM UNIT # TABLE GOES IN SAME SEG W/ FORTRAN I/O

SEG START-STOP:

IN DAM.PLCXQT . PLTCLASS INITIALIZATION ROUTINE
IN DAM.PLCXI . PLTCLASS TERMINATION ROUTINE

SEG SPECIFY*.START-STOP

IN DAM.KMOCLE . CLEAR WARNINGS/ERRORS
IN DAM.KMDCOP . GET/CHECK NUMBER OF OUTPUT COPIES
IN DAM.KMDDEN . GET/CHECK DENSITY LIMITS
IN DAM.KMDEXP . EXPLAIN PROGRAM/COMMAND
IN DAM.KMDMEA . GET/CHECK PAGE HEADINGS
IN DAM.KMDMER . GET/CHECK TRANSVERSE MERCATOR CENTRAL MERIDIAN
IN DAM.KMONEW . PRINT NEWS
IN DAM.KMDNEX . CONDITIONALLY PERFORM NEXT COMMAND
IN DAM.KMDOFF . TURN OFF MODE SWITCHES
IN DAM.KMDON . TURN ON MODE SWITCHES
IN DAM.KMDORI . GET/CHECK WINDOW ORIGIN
IN DAM.KMDPAO . SKIP TO TOP OF NEXT PAGE
IN DAM.KMDPLO . GET/CHECK PLOTTER SPECIFICATIONS
IN DAM.KMDREN . RENUMBER (GET/CHECK NEW WINDOW SEQUENCE NUMBER)
IN DAM.KMOSCA . GET/CHECK WINDOW SCALE
IN DAM.KMOSYM . GET/CHECK SYMBOLS
IN DAM.KMOTIC . GET/CHECK TICK UNITS/INTERVALS
IN DAM.KMOTIM . PRINT CLOCK TIME & CHARGE TIME
IN DAM.KMDWIN . GET/CHECK WINDOW ENVELOPE/VERTICES
IN DAM.KMXXX . MACRO COMMANDS
IN DAM.KMDZON . GET/CHECK UTM PROJECTION ZONE

SEG MAPOUT*.START-STOP

IN DAM.PLCMAP . MAP RADIANCE/DENSITY/CLASS (PHASE 0)

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PLTCLASS-NAP/VIRTUAL
001

IN DAM.PLTCLASS/VIRTUAL


```

      INTEGER KASE          & MODIFIED I-C-E OF FIRST CHAR OF COMMAND
C
C
C PROCEDURE
C -----
C
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
      CALL TRACE
      CALL NANSUB          & CALL TO NULSUB DOES NOTHING
C
C
C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C
      KOMD=' NUL'        & IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
      CALL READSLSSTAT.  & ' ' & FILL BUFFER. BLANK CUE MESSAGE
      IF(SLSSTAT.NE.' ') KOMD='EOF5'
      IF(KOMD.NE.'EOF5') CALL GETSAL(KOMD.(3), NULCST) & GET 3 ALPHA CHARS
C
C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C
      KASE=ICE(KOMD)-ICE('A')+1      & A TO Z = 1 TO 26
C
C
C CASE STATEMENT ON MODIFIED I-C-E OF COMMAND'S FIRST CHARACTER
C
      IF((KASE.LT.1).OR.(KASE.GT.26)) KASE=27      & NOT ALPHA
      GO TO 1
      0 401.402.403.404.405.406.407.408.409.410.
      1 411.412.413.414.415.416.417.418.419.420.
      2 421.422.423.424.425.426.427)
      & .KASE
C
C
C DETERMINE COMMAND. PERFORM COMMAND. CHANGE KOMD TO BLANK
C
      401 CONTINUE &... A
      402 CONTINUE &... B
      GO TO 800
C
      403 CONTINUE &... C
      IF(KOMD.EQ.'CLE') CALL KMCLE(KOMD)      & CLEAR
      IF(KOMD.EQ.'COP') CALL KMCCOP(KOMD)    & CCPIES
      GO TO 800
C
      404 CONTINUE &... D
      IF(KOMD.EQ.'DEN') CALL KMDDEN(KOMD)    & DENSITY
      GO TO 800
C
      405 CONTINUE &... E
      IF(KOMD.EQ.'EOF5') CALL PLCEXT(KOMD)   & END-OF-FILE CAUSES EXIT
      IF(KOMD.EQ.'EXI') CALL PLCEXT(KOMD)   & EXIT
      IF(KOMD.EQ.'EXP') CALL KMDEXP(KOMD)   & EXPLAIN
      GO TO 800
  
```

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PLC000
003

C
406 CONTINUE 3*** F
407 CONTINUE 3*** G
GO TO 800

C
408 CONTINUE 3*** H
IF(KOMD.EQ.'HEA') CALL KMDHEA(KOMD) 3 HEADING
GO TO 800

C
409 CONTINUE 3*** I
410 CONTINUE 3*** J
411 CONTINUE 3*** K
412 CONTINUE 3*** L
GO TO 800

C
413 CONTINUE 3*** M
IF(KOMD.EQ.'MAP') CALL PLCHAP(KOMD) 3 MAP
IF(KOMD.EQ.'MER') CALL KDMER(KOMD) 3 MERIDIAN
GO TO 800

C
414 CONTINUE 3*** N
IF(KOMD.EQ.'NEW') CALL KMDNEW(KOMD) 3 NEWS
IF(KOMD.EQ.'NEX') CALL KMDNEX(KOMD) 3 NEXT
GO TO 800

C
415 CONTINUE 3*** O
IF(KOMD.EQ.'OFF') CALL KMDOFF(KOMD) 3 OFF
IF(KOMD.EQ.'ON ') CALL KMDON(KOMD) 3 ON
IF(KOMD.EQ.'ORI') CALL KMDORI(KOMD) 3 ORIGIN
GO TO 800

C
416 CONTINUE 3*** P
IF(KOMD.EQ.'PAG') CALL KMDPAG(KOMD) 3 PAGE
IF(KOMD.EQ.'PLO') CALL KMDPLO(KOMD) 3 PLOTTER
GO TO 800

C
417 CONTINUE 3*** Q
GO TO 800

C
418 CONTINUE 3*** R
IF(KOMD.EQ.'RAD') CALL KMDRAD(KOMD) 3 RADIANCE
IF(KOMD.EQ.'REN') CALL KMDREN(KOMD) 3 RENUMBER
GO TO 800

C
419 CONTINUE 3*** S
IF(KOMD.EQ.'SCA') CALL KMDSCA(KOMD) 3 SCALE
IF(KOMD.EQ.'SYM') CALL KMDSYM(KOMD) 3 SYMBOLS
GO TO 800

C
420 CONTINUE 3*** T
IF(KOMD.EQ.'TIC') CALL KMDTIC(KOMD) 3 TICKS
IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD) 3 TIME
GO TO 800

C
421 CONTINUE 3*** U
422 CONTINUE 3*** V

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PLC000
004

00 TO 800

C

423 CONTINUE 3*** M
IF(KOMD.EQ.'WIN') CALL KMDWIN(KOMD) 3 WINDOW
00 TO 800

C

424 CONTINUE 3*** X
425 CONTINUE 3*** Y
00 TO 800

C

426 CONTINUE 3*** Z
IF(KOMD.EQ.'ZON') CALL KMDZON(KOMD) 3 ZONE
00 TO 800

C

427 CONTINUE 3*** NOT ALPHABETIC
IF(KOMD.EQ.'3AD') CALL KMD3AD(KOMD) 3 3ADD
IF(KOMD.EQ.'SAD') CALL KMD3AD(KOMD) 3 SADD
IF(KOMD.EQ.'SAS') CALL KMD3AS(KOMD) 3 SASO
IF(KOMD.EQ.'SBR') CALL KMD3BR(KOMD) 3 SBRKPT
IF(KOMD.EQ.'SFR') CALL KMD3FR(KOMD) 3 \$FREE
IF(KOMD.EQ.'SLO') CALL KMD3LO(KOMD) 3 SLOO

C

C

C

C

IF COMMAND WAS NOT FOUND. TRY MACRO-COMMAND

C

800 IF(KOMD.NE.' ') KOMD='PLC-' 3 1ST 3 CHARS OF PROG NAME PLUS '-'
IF(KOMD.NE.' ') CALL KMDXXX(KOMD) 3 MACRO COMMAND HANDLER

C

C

C

C

COMMAND IS INVALID IF STILL NOT FOUND

C

IF(KOMD.NE.' ') CALL WARN5('INVALID COMMAND ---')

C

C

C

C

FORCE ALL FORTRAN I/O ROUTINES INTO SAME SEG AS PLC000 (NEVER PERFORMED)

C

IF(KOMD.EQ.'JUNK') READ(895.895) KOMD
895 FORMAT(IX)

C

C

C

C

RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY

C

RETURN
END

**DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**PLC129
001**

(NOT IMPLEMENTED)

**QAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**PLCXXI
001**

(NOT IMPLEMENTED)

C LOCAL DECLARATIONS

C -----

C
C INTEGER NSUBM & MAXIMUM NUMBER OF SUBWINDOWS
C INTEGER ITEM & TEMPORARY INTEGER

C
C
C PROCEDURE

C -----

C
C
C CALL TRACE

C
C
C GET/CHECK NUMBER OF SUB-WINDOWS

C
C NSUBW=0
C CALL GETSIN(NSUBW, 1.500, 'BAD NUMBER OF SUB-WINDOWS --')
C IF(NCFIRM.NE.0) WRITE(6,115) NSUBW
C 115 FORMAT(' MAP, .13.' SUB-WINDOWS')

C
C
C CALIBRATE/CHECK SPECIFICATIONS

C
C IF(NWNDOW.EQ.0) CALL WARN\$('INVALID DEFAULT COMMAND --')
C CALL GETSIN(ITEMP, +1.-1, 'EXTRA MAP SPECIFICATION --')
C CALL CALSYL
C CALL CALSCA(FLOAT(LINCH),FLOAT(KINCH)) &-&-&-&- NEEDS REVISION &-&
C CALL CALWIN(0.25)
C IF(CHECK.NE.0) GO TO 900

C
C
C CHECK IF ANY DATA LIES WITHIN OUTPUT WINDOW

C
C IF((MSAOWW(WLIN,WMAX).GE.MSAIWW(WLIN,WMAX)).OR.
C & (MSAOWW(WLIN,WMAX).LE.MSAIWW(WLIN,WMIN)).OR.
C & (MSAOWW(WSAM,WMIN).GE.MSAIWW(WSAM,WMAX)).OR.
C & (MSAOWW(WSAM,WMAX).LE.MSAIWW(WSAM,WMIN))) CALL MDWARN(
C ' NO MSS DATA WITHIN WINDOW')

C
C
C OPEN PLOT FILE IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER

C
C IF(NDTOTL.NE.0) GO TO 900
C IF(NWNDOW.LT.0) CALL OPENPL & OPEN PLOT FILE BEFORE 1ST WINDOW
C NWNDOW=ABS(NWNDOW)
C NPAGE=0

C
C
C GENERATE WINDOW MAP

C
C IF(NSUBW.NE.0) GO TO 400
C CALL MAPLOT
C GO TO 900

C
C

C GENERATE GEOGRAPHIC SUBWINDOW MAPS

C

```
400 IF((KSYOWH(WTIC).NE.'DEG').AND.  
      & (KSYOWH(WTIC).NE.'MIN')) GO TO 800      & NOT GEOGRAPHIC  
      CALL SUBWIN( GEDOWN.NSUBW.MAPLOT)  
      GO TO 900
```

C

C GENERATE UTM SUBWINDOW MAPS

C

```
600 IF((KSYOWH(WTIC).NE.'KM ').AND.  
      & (KSYOWH(WTIC).NE.'MET')) GO TO 800      & NOT UTM  
      CALL SUBWIN( UTMOWH.NSUBW.MAPLOT)  
      GO TO 900
```

C

C INVALID TICK INTERVAL FOR GENERATING SUBWINDOW MAPS

C

```
800 CALL MDWARN( 'SUBWINDOWS NOT ALLOWED FOR CURRENT PRIMARY TICKS')
```

C

C ANY DIAGNOSTICS???

C

```
900 IF(INDFATL.EQ.0) GO TO 920  
      CALL MDNOTE( 'PREVIOUS FATAL ERRORS -- NO MAP GENERATED')  
      GO TO 990
```

```
920 IF(NDWARN.EQ.0) GO TO 990  
      CALL MDNOTE( 'PREVIOUS WARNINGS -- NO MAP GENERATED')  
      IF(MBATCH.EQ.0) WRITE(6,925)
```

```
925      FORMAT(4X,'**TRY AGAIN!')  
      CALL MDCLR( NULCST)
```

```
990 WRITE(6,995)
```

```
995 FORMAT('0')
```

```
KOMD=''
```

```
RETURN
```

```
END
```

```
      SUBROUTINE PLCXQT  8  INITIALIZATION ROUTINE FOR PLTCLASS
-----
C
C (R E HARVESON)
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
-----
C
C      PSTART
C      NVIATO
C      ERCSF
C      OPN12N
C      CLSHD0
C
C      INCLUDE KOMXQT.LIST      8  COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
      EXTERNAL PLC000.NULSUB
C
C
C IDENTIFY PROGRAM
C
C      CALL PSTART(  'DAM PLTCLASS(7903)')
C
C ON RETURN. CALL PLC000 TO GET DEFAULT/USER COMMANDS
C
C      CALL NVIATO(  PLC000.NULSUB)
C
C OPEN DETECTION FILES AND IDENTIFY ERTS SCENE
C
C      CALL OPN12N
C      CALL CLSHD0(  8)
C
C
C QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE
C
C 300 CALL SYSADD(LOCFIL,  'MACDAM','DEF-PLTCLASS',. )
      IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL,  'DAM','DEF-PLTCLASS',. )
      IF(LOCFIL.LE.0) CALL M0FATL(  'NO DEFAULT COMMANDS')
C
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
      RETURN
      END
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

ITICPL
001

(NOT IMPLEMENTED)

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

HTICPL
001

(NOT IMPLEMENTED)

**DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**NITMPL
001**

(NOT IMPLEMENTED)

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

OPLPLC
001

(NOT IMPLEMENTED)

**DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**REOPLC
001**

(NOT IMPLEMENTED)

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

FLHCLASS
001

PROGRAM FLHCLASS NOT IMPLEMENTED

NRULIN=20 & MAX NUMBER OF RUNS LISTED FOR NON-PRIVILEGED USER
CALL ERCSF(ISTAT,'&ADD DAN.DEF-STATUS .')

C
C
C

READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)

C

300 KOMD='

CALL READS(LSSTAT, MSOCUE(NCUE)) &

IF(LSSTAT.NE.' ') KOMD='EXIT'

IF(KOMD.EQ.' ') CALL GETSKN(KOMD,(8), NULCST) & GET 8 ALPHA CHARS

C

C

CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT

C

360 KASE=ICE(KOMD)-ICE('A')+1 & A TO Z = 1 TO 26

C

C

CASE STATEMENT ON MODIFIED I-C-E OF COMMAND'S FIRST CHARACTER

C

IF((KASE.LT.1).OR.(KASE.GT.26)) KASE=27 & NOT ALPHA

GO TO 1

0 401,402,403,404,405,406,407,408,409,410,

1 411,412,413,414,415,416,417,418,419,420,

2 421,422,423,424,425,426,427)

6 .KASE

C

C

DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOMD TO BLANK

C

401 CONTINUE 3*** A

402 CONTINUE 3*** B

403 CONTINUE 3*** C

404 CONTINUE 3*** D

GO TO 500

C

405 CONTINUE 3*** E

IF((KOMD.EQ.'EXIT').OR.

& (KOMD.EQ.'EXI')) CALL PSTOP(NULCST)

IF((KOMD.EQ.'EXPLAI').OR.

& (KOMD.EQ.'EXPLA').OR.

& (KOMD.EQ.'EXPL').OR.

& (KOMD.EQ.'EXP')) CALL KMDEXP(KOMD)

GO TO 500

C

406 CONTINUE 3*** F

407 CONTINUE 3*** G

408 CONTINUE 3*** H

409 CONTINUE 3*** I

410 CONTINUE 3*** J

GO TO 500

C

411 CONTINUE 3*** K

IF(KOMD.EQ.'KEY') CALL STAKEY(KOMD)

GO TO 500

C

412 CONTINUE 3*** L

DAN PACKAGE APPENDIX L
 MAIN PROGRAMS/ROUTINES

STATUS
 003

```

413 CONTINUE 3*** M
    GO TO 500
C
414 CONTINUE 3*** N
    IF(KOMD.EQ.'NEWS') .OR.
    & (KOMD.EQ.'NEW') CALL KMDNEW(KOMD)
    IF(KOMD.EQ.'NEXT') .OR.
    & (KOMD.EQ.'NEX') CALL KMDNEX(KOMD)
    GO TO 500
C
    CONTINUE 3*** O
    IF(KOMD.EQ.'OFF') NCUE=1      & ASK FOR RUNID AFTER DEFAULT COMMANDS
    IF(KOMD.EQ.'OFF') CALL KMOFF(KOMD)
    IF(KOMD.EQ.'ON') CALL KMDON(KOMD)
    GO TO 500
C
416 CONTINUE 3*** P
417 CONTINUE 3*** Q
418 CONTINUE 3*** R
419 CONTINUE 3*** S
420 CONTINUE 3*** T
421 CONTINUE 3*** U
422 CONTINUE 3*** V
423 CONTINUE 3*** W
424 CONTINUE 3*** X
425 CONTINUE 3*** Y
426 CONTINUE 3*** Z
    GO TO 500
C
427 CONTINUE 3*** NOT ALPHABETIC
    IF(KOMD.EQ.'') GO TO 300      & IGNORE BLANKS
    GO TO 500
C
C
C LIST RUN SUMMARY
C
500 IF(KOMD.EQ.'') GO TO 300      & COMMAND ALREADY PROCESSED
    NRUNS=0
    NTRNSP=0
    CALL GETRUN
    NCUE=11      & ASK FOR RUNID OR INDEX
    GO TO 300
C
C
C LIST PROGRAM EXECUTION SUMMARY
C
600 CALL UNGETS
    LINDEX=999999
    CALL GETSIN(LINDEX, 4,9000,NULCST)
620 IF(LINDEX.EQ.999999) GO TO 500      & NOT A VALID INDEX. MUST BE RUNID!!
    IF(LGHMAX.NE.0) GO TO 640
    NCUE=4      & ASK FOR RUNID. NOT INDEX!
    GO TO 300
640 CALL GETXQT
    IF(LINDEX.EQ.0) GO TO 300
    CALL READ5(LSSTAT, ' ')
  
```

```

IF(LSSTAT.EQ.'EOF') CALL PSTOP(NULCST)
CALL GETSKH(KOMD,(6), NULCST)
IF(KOMD.EQ.' ') GO TO 840      & MONITOR ACTIVE PROGRAM
GO TO 380
  
```

C
C
C
C
C
C
C

```

SUBROUTINE STAKEY(KOMD)
CALL GETSKH(KEY,(6), NULCST)
CALL VALKEY(KEY)
NACCT=21      & PRINT ACCOUNT NUMBERS
NRULIM=99999  & ALLGW ANY NUMBER OF RUNS TO BE LISTED
WRITE(6,105)
105 FORMAT(' '*DESTROY*SECURITY*KEY*')
KOMD=' '
RETURN
  
```

C
C
C
C
C
C

SUBROUTINE GETRUN

C
C
C
C
C

```

1111112222233333444445555566666
INTEGER JUSER(6) /*JUSE READSX..SYSS*READSXNNNNNN. . .*/
INTEGER JASOR(3) /*ASO.A READSX. . .*/
INTEGER JFREER(3)/*FREE READSX. . .*/
INCLUDE FACBIT.LIST
  
```

C
C
C
C

COUNT TRANSPARENT (//) RUNID CHARACTERS

```

DO 150 NCHAR=1,6
CALL GETCHR(KHAR, KOMD,NCHAR)
IF(KHAR.EQ.' ') NTRNSP=NTRNSP+1
150 CONTINUE
NRUMAX=NTRNSP+2
CALL GETSIN(NRUMAX, .1,NRULIM,'BAD MAXIMUM RUN NUMBER --')
CALL GETSIN(ITEMP, +1,-1,'EXTRA RUN SPECIFICATION --')
IF(NTRNSP.NE.0) GO TO 170      & ONE OR MORE TRANSPARENT CHARS
  
```

C
C
C
C

CHECK RUNID FOR INVALID CHARACTERS

```

DO 160 NCHAR=1,6
CALL GETICE(NICE, KOMD,NCHAR)      & GET INTEGER-CHARACTER-EQUIVALENT
IF(
& (NICE.LT.ICE('A').OR.NICE.GT.ICE('Z')) .AND.
& (NICE.LT.ICE('0').OR.NICE.GT.ICE('9'))
& ) GO TO 170      & INVALID CHARACTER
  
```



```

160 CONTINUE
C
C
C CHECK IF QUEUED READSX CARD INPUT FILE EXISTS FOR SPECIFIED RUNID
C
      JUSER(5)=KOMD      & RUNID
      CALL ERCSF(ISTAT,JUSER)
      CALL ERCSF(ISTAT,JASOR)
      IF(ACCEPT(ISTAT).EQ.0) GO TO 170      & NO READSX QUEUED CARD INPUT FILE
C
C
C IDENTIFY QUEUED RUN AS BACKLOG OR ACTIVE
C
      LORNEW=-999999
      LPCT(1)=KOMD      & RUNID
      FLD(06,6,LPCT(25))=6      & 'BATCH'
      LPCT(15)='889888'      & DATE = 00/00/88 TIME = 00:00
      IF(OTHRUN(ISTAT).EQ.0) GO TO 165      & NOT YET BEING USED
      LPCT(18)='(ACTIV'
      LPCT(19)='      'E)'
      GO TO 167
165 LPCT(18)='(BACKL'
      LPCT(19)='      '00)'
167 CALL ERCSF(ISTAT,JFREER)
      CALL PRTRUN
C
C
C READ LOG FILE HEADER
C
170 IOSIZE(LGPKT)=5
      IOADDR(LGPKT)=LOC(LGMHDR)
      IOSECT(LGPKT)=0
      IOFUNC(LGPKT)='BK'      & READ
      CALL ERLOW(LGPKT)
      LORNEW=LGRHI
C
C
C CHAIN READ RUN HEADERS IN REVERSE ORDER
C
      IOSIZE(LGPKT)=28
      IOADDR(LGPKT)=LOC(LGROLD)
200 IOSECT(LGPKT)=LORNEW
      IOFUNC(LGPKT)='BK'      & READ
      CALL ERLOW(LGPKT)
      IF(MATCHR(KOMD,LPCT(1)).GE.(6-NTRNSP)) CALL PRTRUN
      IF(NRUNS.GE.NRUMAX) GO TO 900
      IF(LGROLD.LT.+4) GO TO 900
      IF((LORNEW.GE.LGPLO).AND.
1 (LGROLD.LE.LGPLO)) GO TO 900
      LORNEW=LGROLD
      GO TO 200
C
C
C CHECK IF ANY SPECIFIED RUNS WERE FOUND
C
900 IF(NRUNS.NE.0) GO TO 990

```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

STATUS
006

```
      MSNONE(2)=KOHM
      CALL MDHARN( MSNONE)
990  RETURN
C
C
C
C
C
C
      FUNCTION MATCHR(JWD1,JWD2)
C
      MATCHR=0
      DO 150 NBIT=0,30,6
      IF(FLD(NBIT,6,JWD1).EQ.FLD(NBIT,6,JWD2)) MATCHR=MATCHR+1
150  CONTINUE
      RETURN
C
C
C
C
C
C
      SUBROUTINE PRTRUN
C
      DIMENSION MODE(6)
      DATA MODE/'HIEXEC','RLTIME','LOEXEC','DEMAND','DEADLN','BATCH'/
C
      IF(NRUNS.EQ.0) WRITE(6,135)
135  FORMAT('0/' INDEX RUNID  MODE
1  'DATE      TIME
2  'QUALIFIER  ACCOUNT'/IX)
      LTOP=FLD(06,6,LPCT(25))
      LMON=FLD(00,6,LPCT(15))
      LDAY=FLD(06,6,LPCT(15))
      LYR =FLD(12,6,LPCT(15))+64
      LHR = FLD(19,18,LPCT(15))/3600
      LMIN=(FLD(18,18,LPCT(15))-3600*LHR)/60
      WRITE(6,155) LGRNEW,LPCT(1),MODE(LTOP),
1  LMON,LDAY,LYR,LHR,LMIN,
2  (LPCT(N),N=18,NACCT)
155  FORMAT(16,2X,A6,2X,A6,2X,
1  J2,'',J2,'',J2,2X,J2,'',J2,2X,
2  2A6,2X,2A6)
      NRUNS=NRUNS+1
      RETURN
C
C
C
C
C
C
      SUBROUTINE GETXQT
C
      DIMENSION LQUAL(2),MDAYS(12)
C
      MONTH: 01 02 03 04 05 06 07 08 09 10 11 12
```

DATA MDAYS/31.29.31.30.31.30.31.31.30.31.30.31/

C
 C
 C
 C

READ LOG SECTORS FOR FIRST PROGRAM EXECUTION OF RUN

```

110 LSECTR=LINDEX
    LINDEX=0
    IF(LSECTR.LT.+4) GO TO 810
    IF(LSECTR.GT.LOMAX) GO TO 810
    IF(MOD(LSECTR,4).NE.0) GO TO 810
    IOSIZE(LGPKT)=100      & 3 SECTORS & FIRST 16 WORDS OF NEXT SECTOR
    IOSECT(LGPKT)=LSECTR
    IOFUNC(LGPKT)='BK'    & READ
    CALL ERLOW(LGPKT)
    IF(IOSTAT(LGPKT).NE.0) GO TO 810
    IF(MATCHR(LPCT(1),LPCT(2)).LT.5) GO TO 810 & ORIG/GEN RUNID NOT SIMILAR
    LTP=FLD(06.6,LPCT(25))
    IF(KOMD.EQ.' ') GO TO 310      & MONITOR ACTIVE PROGRAM
    IF(LGOLD.EQ.0) CALL WARN5('PARTIAL RUN INDEX --')
    LRORIG=LPCT(1)
    LRGEN =LPCT(2)
    LMDYS=LPCT(15)
    LMON=FLD(00.6,LPCT(15))
    LDAY=FLD(06.6,LPCT(15))
    LYR =FLD(12.6,LPCT(15))+64
    MDAYS(2)=28
    IF(MOD(LYR,4).EQ.0) MDAYS(2)=29      & LEAP YEAR!
    LQUAL(1)=LPCT(18)
    LQUAL(2)=LPCT(19)
    WRITE(6,125) LSECTR,LRORIG,LRGEN,LQUAL
125 FORMAT('07' ('J5.' '.A6.'/''.A6.' '''.2A6/1X.4('-----'))
    GO TO 310
  
```

C
 C
 C
 C

READ LOG SECTORS FOR SUBSEQUENT PROGRAM EXECUTIONS

```

200 IOSECT(LGPKT)=LSECTR
    IOFUNC(LGPKT)='BK'    & READ
    CALL ERLOW(LGPKT)
    IF(IOSTAT(LGPKT).NE.0) GO TO 900
    IF(MATCHR(LPCT(1),LPCT(2)).LT.5) GO TO 900
    IF((FLD(6.6,LPCT(15)).NE.LDAY).AND.
    I (FLD(6.6,LPCT(15))-MOD(LDAY,MDAYS(LMON))).NE.+1) GO TO 900
    IF(LPCT(2).NE.LPGEN) GO TO 400
    IF(LPCT(15).NE.LMDYS) GO TO 400
  
```

C
 C
 C
 C

WRITE QUALIFIER IF CHANGED

```

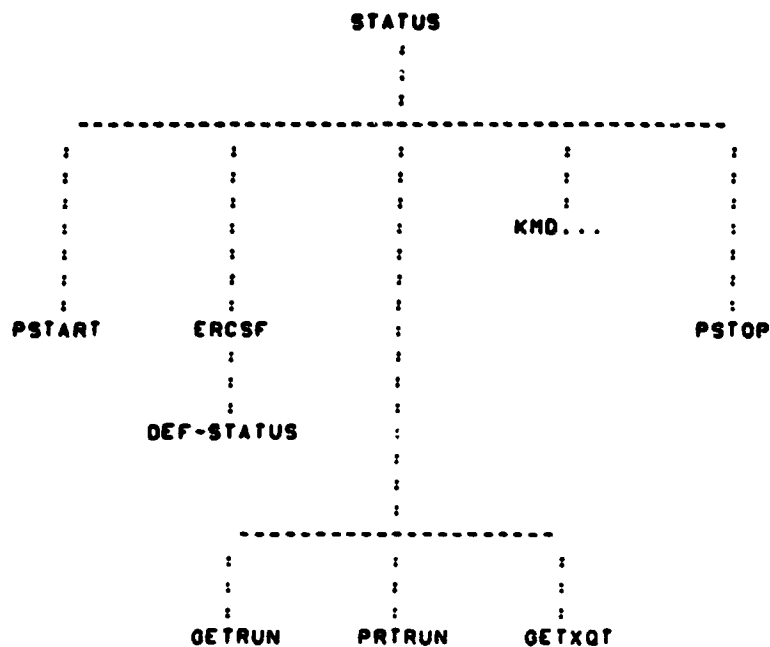
300 LINDEX=0
    IF((LQUAL(1).EQ.LPCT(18)).AND.
    I (LQUAL(2).EQ.LPCT(19))) GO TO 310
    LQUAL(1)=LPCT(18)
    LQUAL(2)=LPCT(19)
    WRITE(6,305) LQUAL
  
```

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

STATUS
000

```
305 FORMAT(' QUALIFIER CHANGED TO '.2A6/1X.4('-----'))
C
C
C WRITE PROGRAM ID/DATE/TIME & DIAGNOSTIC/TERMINATION LOG ENTRIES
C
310 WRITE(6,310) LPIDT
315 FORMAT(1X,15A6)
   IF(LDIAON(1).NE.' ') WRITE(6,325) LDIAON
325 FORMAT(2X,R2,14A8)
   WRITE(6,335) LTERM
335 FORMAT(1X,14A6/1X.4('-----'))
C
C
C CHECK PROGRAM TERMINATION STATUS
C
   IF((LTERM(1).EQ.' NORMA') .OR.
& (LTERM(1).EQ.' ERRO')) GO TO 400
   IF(LTERM(1).NE.' ABOR') GO TO 380
   IF(LTYP.EQ.4) GO TO 400      & IN DEMAND ABORT DOES NOT KILL RUN
   GO TO 900
C
C
C INCOMPLETE PROGRAM EXECUTION -- SAVE SECTOR POINTER
C
380 LINDEX=LSECTR
C
C
C TEST AND INCREMENT LOG INDEX
C
400 IF(LSECTR.EQ.LOPHI) GO TO 900
   LSECTR=MOD(LSECTR,LOMAX)+4
   GO TO 200
C
C
C FLAG INVALID INDEX
C
810 CALL HDWARN('INVALID INDEX---')
   WRITE(6,815) LSECTR
815 FORMAT(6X,16)
C
900 RETURN
   END
```

STATUS HIERARCHY



PROGRAM STATUS/VIRTUAL

HISTORY

E M SCHLOSSER	LEC	08/02/74	ORIGINAL CODE
E M SCHLOSSER	LEC	11/08/79	MAP.FZ(IN): NO 'N' IN DEMAND

METHOD

CONSTRUCT MAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT SXQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE MAP & SXQT COMMANDS TO TEMPORARY FILE 20.
BADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 8-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES.
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS	3	FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IOWS	3	INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS	3	TERMINATE PROGRAM EXECUTION
DAM.STATUS-MAP	3	SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAOPT	3	STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS BASO-D & APREP-D.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

1	2	=	REAL TIME
1	3	=	LOW EXEC
1	4	=	DEMAND
1	5	=	DEADLINE BATCH
1	6	=	BATCH

(SXQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER A5 IN

(MASTER BIT NOTATION.)

LOCAL DECLARATIONS

```

      AXRS
S(00) . D-BANK
SSSH  FORM          6.6.6.18
      1111122222333333444445555566666677777788888899999
LABSDF SSSH          050.1.'F'.0 . LABEL. 1 WD. FORTRAN. FIELDATA
      *SDF*
LABIMO SSSH          000.9.0.0 . DATA. 9 WDS. . FIELDATA
MAPSDF *BXQTS: MAP.FZN DAM.STATUS-MAP.STATUS . :BXQTS*
MAPIMO SSSH          000.9.0.0
ADDSDS SSSH          000.9.0.0
ADDIMO *BXQTS: ADD CAM.SYS-MAPOPT . :BXQTS*
XQTSDF SSSH          000.9.0.0
XQTIMO *BXQTS: XQT.1 . STATUS . :BXQTS*
EOFSDF - 0 . END-OF-FILE STOP WORD
PF     FORM          12.6.18
CSFASO *BASO.T 20. . .
CSFADD *ADD 20. . .
SAVREG RES          1
IOPKT  1500         *20*.WS 33.LABSDF.'0' 0
  
```

PROCEDURE

```

S(01) . I-BANK
STATUS: LA.U        A0. . . . . A0 := .
      TNE.U        A4.4 . . . . . SKIP NEXT INST IF A4<>4 (NOT DEMAND)
      SA.S2       A0.MAPIMO*2 . DEMAND! BLANK OUT N OPTION
      LA          A0.(CSFASO) . ADDRESS OF BASO IMAGE
      ER          CSFS . DO IT
      SA          A0.SAVREG . STORE 8
      PSRINT      (PF 2.1.SAVREG) . PRINT BASO STATUS
      GETOPT      . LOAD OPT LTRS INTO A2.A3.A4
PUTOPT  DS        A2.XQTIMO*2 . STORE OPTION LETTERS INTO BXQT IMAGE
      SA          A4.XQTIMO*4 . (3 WORDS -- MAX 18 OPT LETTERS)
WRITE   LA        A0.(IOPKT) . ADDRESS OF I/O PACKET
      ER          IOWS . WRITE SDF IMAGES TO 20.
ADD     LA        A0.(CSFADD) . ADDRESS OF 3ADD IMAGE
      ER          CSFS . DO IT
      ER          EXITS
      END        STATUS
  
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

STATUS-MAP
001

IN DAN.STATUS/..NTABS/DAN..SYS-BLOCK
LIB DAN.

DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

STATUS-MAP/VIRTUAL
001

IN DAH.STATUS/VIRTUAL

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DITCOP
001

```
C
C
C
C   PROGRAM DITCOP
C   -----
C
C   (J C POOLEY)
C
C   THIS PROGRAM COPIES *DAMDET-11-4) DISC DETECTION FILES TO
C   TAPE.
C
C   INPUT CONSISTS OF *DAMDET-11-4) DENSITY DISC FILES.
C   OUTPUT CONSISTS OF ONE TO FOUR FILES ON A SINGLE REEL OF TAPE.
C   THE DISC FORMAT IS DOCUMENTED IN DAM PACKAGE CLASSIFY.
C   THIS TAPE FILE IS IN UNIVAC EXEC 8 3COPY FORMAT.
C
C
C   INCLUDE KOMXQT
C   INCLUDE KOMLOO
C   INCLUDE KOMLUS
C   INCLUDE KOMNER
C   INCLUDE KOMFIT
C
C
C   EXTERNAL DIT000.DITXQT
C
C   CALL NVIATO( DIT000.DITXQT) 8 FIRST CALL IS VIA DIT000 TO DITXQT
100 CALL VIATO
    GO TO 100
    END
```

PROGRAM DITCOP/VIRTUAL

HISTORY

E H SCHLOSSER	LEC	07/07/78	ORIGINAL CODE
E H SCHLOSSER	LEC	11/08/79	MAP.FZ(N): NO 'N' IN DEMAND

METHOD

CONSTRUCT MAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFs.
CONSTRUCT XQT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFs.
WRITE MAP & XQT COMMANDS TO TEMPORARY FILE 20.
ADD TEMPORARY FILE 20. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-8 OPERATING SYSTEM USING 6-BIT FIELDATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS	3	FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IOWS	3	INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS	3	TERMINATE PROGRAM EXECUTION
DAM.DITCOP-MAP	3	SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAPOPT	3	STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS BASO-D & APREP-D.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A4 AS FOLLOWS:

(2 = REAL TIME
(3 = LOW EXEC
(4 = DEMAND
(5 = DEADLINE BATCH
(6 = BATCH

(XQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER A5 IN

C-6

ORIGINAL PAGE 23
OF FOUR QUALITY

(MASTER BIT NOTATION.

LOCAL DECLARATIONS

```

      AXRS
$ (00) . D-BANK
SSSH      FORM      6.6.6.18
           111111222222333333444444555555666666777777888888999999
LABSDF    SSSH      090.1.'F'.0      . LABEL. 1 NO. FORTRAN. FIELDATA
LABIMG    'SDF'
MAPSDF    SSSH      000.9.0.0      . DATA. 9 WDS. . FIELDATA
MAPIMG    'BXQTS: MAP.FZN DAH.DITCOP-MAP.DITCOP      . :BXQTS'
ADDSDF    SSSH      000.9.0.0
ADDIMG    'BXQTS: ADD      DAH.SYS-MAPOPT      . :BXQTS'
XQTSDF    SSSH      000.9.0.0
XQTING    'BXQTS: XQT.1      DITCOP      . :BXQTS'
EOFSDF    -      0      . END-OF-FILE STOP WORD
PF        FORM      12.6.18
CSFASG    'ASG.T 20. . .
CSFADD    'ADD 20. . .
SAVREG    RES      1
IOPKT     ISOD     '20'.WS 33.LABSDF.'0' 0
  
```

PROCEDURE

```

$ (01) . I-BANK
DITCOP*   LA.U      A0.'      . A0 := .
          TNE.U     A4.4      . SKIP NEXT INST IF A4<>4 (NOT DEMAND)
          SA.S2     A0.MAPIMG+2 . DEMAND! BLANK OUT N OPTION
          LA        A0.(CSFASG) . ADDRESS OF ASG IMAGE
          ER        CSFS      . DO IT
          SA        A0.SAVREG  . STORE &
          PSRINT    (PF 2.1.SAVREG) . PRINT ASG STATUS
          GETOPT    . LOAD OPT LTRS INTO A2.A3.A4
          PUTOPT   DS      A2.XQTING+2 . STORE OPTION LETTERS INTO BXQT IMAGE
          SA        A4.XQTING+4 . (3 WORDS -- MAX 18 OPT LETTERS)
          WRITE    LA      A0.(IOPKT) . ADDRESS OF I/O PACKET
          ER        IOWS     . WRITE SDF IMAGES TO 20.
          ADD      LA      A0.(CSFADD) . ADDRESS OF ADD IMAGE
          ER        CSFS     . DO IT
          ER        EXITS
          END      DITCOP
  
```

**DAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES**

**DITCOP-NAP
001**

**IN DAH.DITCOP/..NTABS/DAH..SYS-BLOCK
LIB DAH.**

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DITCOP-MAP/VIRTUAL
001

IN DAN.DITCOP/VIRTUAL

ORIGINAL PAGE 2
OF FOUR QUALITY


```

      INTEGER KASE          & MODIFIED I-C-E OF FIRST CHAR OF COMMAND
C
C
C PROCEDURE
C -----
C
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
      CALL TRACE
      CALL NAMSUB          & CALL TO NULSUB DOES NOTHING
C
C
C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C
      KOMD=' NUL'        & IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
      CALL READ5(LSSTAT, NULCST)    & FILL UNIT 5 BUFFER, NO CUE MESSAGE
      IF(LSSTAT.NE.' ') KOMD='EXI'
      IF(KOMD.NE.'EXI') CALL GET5AL(KOMD.(3), NULCST)    & GET 3 ALPHA CHARS
C
C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C
      KASE=ICE(KOMD)-ICE('A')+1    & A TO Z = 1 TO 26
C
C
C CASE STATEMENT ON MODIFIED ICE OF COMMAND'S FIRST CHARACTER
C
      IF((KASE.LT.1).OR.(KASE.GT.26)) KASE=27    & NOT ALPHA
      GO TO(
0 401,402,403,404,405,406,407,408,409,410,
1 411,412,413,414,415,416,417,418,419,420,
2 421,422,423,424,425,426,427)
      & .KASE
C
C
C DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOMD TO BLANK
C
401 CONTINUE &... A
      GO TO 800
C
402 CONTINUE &... B
      GO TO 800
C
403 CONTINUE &... C
      IF(KOMD.EQ.'CLE') CALL KMOCLE(KOMD)    & CLEAR
      GO TO 800
C
404 CONTINUE &... D
      IF(KOMD.EQ.'DUP') CALL DTDUP(KOMD)
      GO TO 800
C
405 CONTINUE &... E
      IF(KOMD.EQ.'EXI') CALL DITEXI(KOMD)    & EXI*
      IF(KOMD.EQ.'EXP') CALL KMDEXP(KOMD)    & EXPLAIN
      GO TO 800

```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

017000
003

```
C
406 CONTINUE 3000 F
407 CONTINUE 3000 G
408 CONTINUE 3000 H
409 CONTINUE 3000 I
410 CONTINUE 3000 J
411 CONTINUE 3000 K
412 CONTINUE 3000 L
413 CONTINUE 3000 M
GO TO 800

C
414 CONTINUE 3000 N
IF(KOMD.EQ.'NEW') CALL KMDNEH(KOMD)      3 NEWS
IF(KOMD.EQ.'NEX') CALL KMDNEX(KOMD)     3 NEXT
GO TO 800

C
415 CONTINUE 3000 O
IF(KOMD.EQ.'OFF') CALL KMDOFF(KOMD)     3 OFF
IF(KOMD.EQ.'ON ') CALL KMDON(KOMD)      3 ON
GO TO 800

C
416 CONTINUE 3000 P
417 CONTINUE 3000 Q
418 CONTINUE 3000 R
GO TO 800

C
419 CONTINUE 3000 S
420 CONTINUE 3000 T
IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD)     3 TIME
421 CONTINUE 3000 U
422 CONTINUE 3000 V
IF(KOMD.EQ.'VER') CALL DITVER(KOMD)
GO TO 800

C
423 CONTINUE 3000 W
424 CONTINUE 3000 X
425 CONTINUE 3000 Y
426 CONTINUE 3000 Z
427 CONTINUE 3000 NOT ALPHABETIC

C
C
C IF COMMAND WAS NOT FOUND. TRY MACRO-COMMAND
C
800 IF(KOMD.NE.' ') KMD='DIT-'      3 1ST 3 CHARS OF PROG NAME PLUS '--'
IF(KOMD.NE.' ') CALL KMDXXX(KOMD)  3 MACRO COMMAND HANDLER

C
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
IF(KOMD.NE.' ') CALL WARNS('INVALID COMMAND --')

C
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
RETURN
END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE DITEXI  & TERMINATION ROUTINE FOR DITCOP
      -----
C
C (E H SCHLOSSER)
C
C      INCLUDE KOMXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C      INCLUDE NULCST.LIST     & DEFINE NULL CHARACTER STRING
C
C      CALL TRACE
C
C      EXTERNAL SUBROUTINES/FUNCTIONS CALLED
C      -----
C
C      PSTOP
C
C      TERMINATE PROGRAM
C
C      IF(INDFATL.NE.0) CALL PABORT(NULCST)
C      CALL PSTOP(NULCST)
C
C      PSTOP DOES NOT RETURN
C
C      END
```


DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DITDUP
002

```
      INCLUDE ASHDEF.LIST      & UNIVAC 1100 ASSEMBLER PARTIAL WORD
      INCLUDE KOHL2N.LIST     & I/O PACKETS FOR DETECTION FILES
      INCLUDE KOHLUS.LIST     & COMMON BUFFER.POINTERS.FLAOS FOR UNIT 8
      INCLUDE FIDEF .LIST     & DEFINES RECORD STRUCTURE

C
C
C LOCAL DECLARATIONS
C -----
C
      INTEGER IDFILE(10)      & ARRAY IN FIDEF FORMAT
      INTEGER
      & JASOT2(4) // 'BASO.T 20. . FILASO  ' /

C
C
C PROCEDURE
C -----
C
      CALL TRACE

C
C ASSIGN SCRATCH FILE 20
C
      CALL ERCSF(NAO.JASOT2)

C
C GET FACILITIES ASSIGNMENT FOR UNIT 2
C
      CALL FLINFO(IDFILE. '2','88')
      IF(IDFILE(FIDEQT).NE.'TAPE')
      & CALL HDFATL( 'TAPE NOT ASSIGNED TO UNIT 2 ')
      IF(INDOTL.NE.0) GO TO 700

C
C WRITE TOC FILE TO TAPE
C
      IOFUNC(LU2PKT)='8C'
      IOSIZE(LU2PKT)=SIZEET
      IOADDR(LU2PKT)=LOC(KOMDET)
      IOWAIT(LU2PKT)=30

C
      CALL ERIOH(LU2PKT)

C
C WRITE END OF FILE TO TAPE
C
      IOFUNC(LU2PKT)='8D'

C
      CALL ERIOH(LU2PKT)

C
C CHECK DATE WORD JENDMY(NCCT) OF DETECTION FILES
C
      DO 800 NCCT=1,4
      IF(JENDMY(NCCT).EQ.' ') GO TO 800

C
C
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DITDUP
003

```
C *DANDET-(NCCT) CATALOGED
C
C   WRITE(20.305) NCCT
C   305 FORMAT('SCOPY.ON *DANDET-'.11.'..2.')
C
C
C END OF LOOP VERIFYING ASSIGNMENT
C
C 600 CONTINUE
C
C
C WRITE MESSAGE TO REWIND TAPE FILE 2.
C
C 700 WRITE(20.705)
C 705 FORMAT('SQGT ERSPTCN'/'0** PLEASE SFREE 2. OR BREWIND 2.'/
C      ('SEOF'))
C
C
C ENDFILE 20 TO DRAIN BUFFER
C
C   ENDFILE 20
C
C
C SADD FILE 20 TO RUNSTREAM
C
C   CALL ERCSF(NAO.'SADD 20. . COPY FILE ')
C
C
C CHECK DIAGNOSTIC COUNTERS
C
C 800 IF(INDHARN.EQ.0) GO TO 820
C   CALL HDNOTE('PREVIOUS WARNINGS -- NO COPYING PERFORMED')
C   IF(INDATCH.EQ.0) WRITE(6.815)
C 815 FORMAT(' ...TRY AGAIN')
C   GO TO 900
C 820 IF(INDFATL.EQ.0) GO TO 850
C   CALL HDNOTE('PREVIOUS FATAL ERRORS -- NO COPYING PERFORMED')
C   GO TO 900
C 850 IF(INDCHECK.EQ.0) GO TO 900
C   CALL HDNOTE('CHECKOUT MODE -- NO COPYING PERFORMED')
C
C
C
C NORMAL RETURN
C
C 900 KOND-'
C   RETURN
C   END
```

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

DITVER
001

SUBROUTINE DITVER(8 VERIFY DAMEN TAPE COPY
& KOND) 8 NORMAL RETURN

C
C
C
C
C
C

(J C POOLEY)

INCLUDE KOMXGT

CALL MONOTE('VERIFY COMMAND NOT YET IMPLEMENTED')
900 KOND= .
RETURN
END

```
      SUBROUTINE DITXQT  0  INITIALIZATION ROUTINE FOR DITCOP
      -----
C
C
C (J C POOLEY)
C
C
C EXTERNAL SUBROUTINES/FUNCTIONS CALLED
C -----
C
C      CLSHOO
C      ERCSF
C      OPNIZH
C      PSTART
C      NVIATO
C
C      INCLUDE KONNER
C      EXTERNAL DIT000.NULSUB
C
C IDENTIFY PROGRAM
C      CALL PSTART('DAN DITCOP(7903) ')
C
C OPEN DENSITY FILES
C      CALL OPNIZH
C
C IDENTIFY ERTS SCENE
C      CALL CLSHOO(6)
C
C ADD DEFAULT COMMANDS
C      300 CALL ERCSF(NAO,'@ADD DAN.DEF-DITCOP ')
C
C ON RETURN, CALL DIT000 TO GET DEFAULT/USER COMMANDS
C      CALL NVIATO(DIT000.NULSUB)
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
      RETURN
      END
```

PREFACE TO APPENDIX M

THE SUBROUTINES IN THIS APPENDIX IMPLEMENT COMMANDS COMMON TO SEVERAL DAM PACKAGE PROGRAMS. THE SUBROUTINE NAMES ARE COMPOSED OF 'KMD' FOLLOWED BY THE FIRST 3 CHARACTERS OF THE COMMAND KEY WORD.

THESE COMMAND SUBROUTINES ARE DESIGNED TO BE CALLED DIRECTLY BY THE CONVERSATIONAL MONITOR FOR EACH PROGRAM, WITH A SINGLE ARGUMENT PROVIDING BOTH INPUT AND OUTPUT. ON ENTRY THIS ARGUMENT MUST CONTAIN THE FIRST 3 CHARACTERS OF THE COMMAND. (KMDPO1 & KMDXXX ARE EXCEPTIONS.) ON RETURN THIS ARGUMENT MUST CONTAIN BLANKS IF THE COMMAND IS VALID (REGARDLESS OF WHETHER THE SPECIFICATIONS ARE CORRECT, OR OF WHETHER THE COMMAND IS ACTUALLY PERFORMED). KMDPO1 & KMDXXX CHECK IF THE COMMAND IS VALID BEFORE DECIDING WHETHER TO RETURN BLANKS. ALL OTHER COMMAND SUBROUTINES WILL ONLY BE CALLED IF THE COMMAND IS VALID, AND THEREFORE MUST ALWAYS RETURN BLANKS.

ALL OTHER INPUTS AND OUTPUTS FOR COMMAND SUBROUTINES ARE PROVIDED BY LABELLED COMMON BLOCKS AND EXTERNAL DEVICE I/O.

EACH COMMAND SUBROUTINE MUST CHECK THE VALUES OF ITS USER SPECIFICATIONS. THE ROUTINE ISSUES DIAGNOSTICS FOR INVALID SPECIFICATION VALUES AND UPDATES VARIABLES IN LABELLED COMMON BLOCKS WITH CORRESPONDING VALID SPECIFICATION VALUES.

IF THE CONFIRM SWITCH IS ON (MCFIRM IN KOMXQT <> 0), A COMMAND SUBROUTINE MUST CONFIRM THE COMMON VALUES OF ALL SPECIFICATION(S) FOR THAT COMMAND (INCLUDING ANY OPTIONAL SPECIFICATION(S) OMITTED BY THE USER). THIS CONFIRMATION SERVES THREE PURPOSES:

1. IT CONFIRMS USER-SUBMITTED SPECIFICATIONS.
2. IT IDENTIFIES CURRENT COMMON VALUES OF OPTIONAL SPECIFICATIONS OMITTED BY THE USER.
3. IT ILLUSTRATES COMMAND SYNTAX.

THE CONFIRMATION MUST INCLUDE THE COMPLETE (UNABREVIATED) KEY WORD FOR THE COMMAND, THE SPECIFICATION VALUE(S), AND THE COMMAS SEPARATING THEM, AND MAY INCLUDE BRIEF EXPLANATORY 'NOISE' WORDS AFTER NUMERIC SPECIFICATIONS. THE CONFIRMATION MUST BE SYNTACTICALLY CORRECT, SUCH THAT IF KEYED IN VERBATIM BY A USER IT WOULD BE ACCEPTABLE.

SEE APPENDIX L FOR COMMAND SUBROUTINES DEDICATED TO INDIVIDUAL PROGRAMS.

SEE APPENDIX D FOR DETAILED SYNTAX OF ALL COMMANDS.

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

APPENDIX-M
001

SET TABS @ 13 & 26

SPRT.SC	DAM.PREFACE-M	. (8005)
SPRT.SC	DAM.APPENDIX-M	.
MSO.N	...ADJ	. ADJUST NETWORK (SEE APPENDIX L)
SPRT.SC	DAM.KMDALI	. ALIGN COORDINATE SYSTEMS
SPRT.SC	DAM.KMDATT	. GET/CHECK PLATFORM ATTITUDE
SPRT.SC	DAM.KMOCEN	. GET/CHECK SCENE CENTER SCAN COORDINATES
SPRT.SC	DAM.KMDCOA	. GET/CHECK RAW/TRANSFORMED CHANNEL(S)
SPRT.SC	DAM.DETCHA	. GET/CHECK DETECTION CHANNEL(S)
SPRT.SC	DAM.KMDCLE	. CLEAR WARNINGS/ERRORS
SPRT.SC	DAM.KMDCOL	. GET/CHECK COLOR(S)
SPRT.SC	DAM.KMDCOP	. GET/CHECK NUMBER OF OUTPUT COPIES
SPRT.SC	DAM.KMDCOU	. GET/CHECK COUNT PER PIXEL
SPRT.SC	DAM.KMDCRO	. CROSSTABULATE
SPRT.SC	DAM.KMDDEN	. GET/CHECK DENSITY LIMITS
MSO.N	...DET	. DETECT (SEE APPENDIX L)
MSO.N	...DIA	. DIAGRAM NETWORK (SEE APPENDIX L)
MSO.N	...DIS	. DISPLAY ON ALPHA-NUMERIC DEVICE (SEE APPENDIX L)
MSO.N	...EXI	. EXIT (SEE APPENDIX L)
SPRT.SC	DAM.KMDEXP	. EXPLAIN PROGRAM/COMMAND
MSO.N	...FAC	. FACTOR (SEE APPENDIX L)
SPRT.SC	DAM.KMDFI	. END IF ... FI BLOCK
SPRT.SC	DAM.KMDGEO	. GET/CHECK SCENE GEOMETRY
SPRT.SC	DAM.KMDHEA	. GET/CHECK PAGE HEADINGS
MSO.N	...HIS	. HISTOGRAM (SEE APPENDIX L)
SPRT.SC	DAM.KMDIF	. BEGIN IF ... FI BLOCK
SPRT.SC	DAM.KMDINT	. GET/CHECK INTENSITY(S)
SPRT.SC	DAM.KMDLIN	. GET/CHECK LINEAR WEIGHTS/GAIN/BIAS
MSO.N	...LIS	. LIST (SEE APPENDIX L)
SPRT.SC	DAM.KMDMER	. MAP (SEE APPENDIX L)
SPRT.SC	DAM.KMDMER	. GET/CHECK TRANSVERSE MERCATOR CENTRAL MERIDIAN
MSO.N	...MOD	. MODEL (SEE APPENDIX L)
SPRT.SC	DAM.KMDNAM	. GET/CHECK MATERIAL NAME
SPRT.SC	DAM.KMDNEW	. PRINT NEWS OF PROGRAM CHANGES
SPRT.SC	DAM.KMDNEX	. SPECIFY CONDITION FOR PERFORMING NEXT COMMAND
SPRT.SC	DAM.KMDOFF	. TURN OFF MODE SWITCH(ES)
SPRT.SC	DAM.KMDON	. TURN ON MODE SWITCH(ES)
SPRT.SC	DAM.KMDORI	. GET/CHECK WINDOW ORIGIN
SPRT.SC	DAM.KMDPAO	. SKIP TO TOP OF NEXT PAGE
MSO.N	...PAR	. PARTITION (SEE APPENDIX L)
SPRT.SC	DAM.KMDPEE	. 'PEEK' INTO LABELLED COMMONS (FOR DEBUGGING)
MSO.N	...PIC	. PICTURE ON COLOR CRT (SEE APPENDIX L)
SPRT.SC	DAM.KMDPLO	. GET/CHECK PLOTTER SPECIFICATIONS
SPRT.SC	DAM.KMDPOI	. GET/CHECK CONTROL/CHECK POINT
SPRT.SC	DAM.KMDPOK	. 'POKE' (CHANGE) LABELLED COMMONS (FOR DEBUGGING)
SPRT.SC	DAM.KMDPOL	. GET/CHECK POLAR GAIN/BIAS
SPRT.SC	DAM.KMDPRI	. GET/CHECK PRINTER SPECIFICATIONS
MSO.N	...PRO	. PROFILE (SEE APPENDIX L)
SPRT.SC	DAM.KMDRAD	. GET/CHECK RADIANCE LIMITS
MSO.N	...RAD	. (SEE ALSO APPENDIX L)
SPRT.SC	DAM.KMDRAN	. GET/CHECK RANKING(S)
SPRT.SC	DAM.KMDREN	. GET/CHECK NEW WINDOW SEQUENCE NUMBER
SPRT.SC	DAM.KMDRES	. GET/CHECK RESAMPLING SPECIFICATIONS
MSO.N	...ROT	. ROTATE (SEE APPENDIX L)
SPRT.SC	DAM.KMDSCA	. GET/CHECK WINDOW SCALE
SPRT.SC	DAM.KMDSCC	. GET/CHECK SCENE NUMBER AND SIZE
SPRT.SC	DAM.KMDSCR	. GET/CHECK SCRIPT

DAM PACKAGE APPENDIX H
COMMAND ROUTINES

APPENDIX-H
002

BPRT.SC DAM.KNOSHA	. GET/CHECK SHARPENING FILTER COEFFICIENTS
BPRT.SC DAM.KNOSIZ	. GET/CHECK SCENE SIZE IN SCAN COORDINATES
BPRT.SC DAM.KNOSPA	. GET/CHECK WINDOW SPACING
BPRT.SC DAM.KNOSPH	. GET/CHECK SPHEROID GEODETTIC PARAMETERS
BPRT.SC DAM.KNOSYM	. GET/CHECK SYMBOL(S)
BPRT.SC DAM.KNOTAB	. TABULATE
BPRT.SC DAM.KNOTIC	. GET/CHECK TICK INTERVALS
BPRT.SC DAM.KNOTIM	. PRINT CLOCK TIME & CHARGE TIME
BPRT.SC DAM.KNOTIN	. GET/CHECK TINT SPECIFICATIONS
BPRT.SC DAM.KNOTOL	. GET/CHECK TOLERANCE
BPRT.SC DAM.KNDWIN	. GET/CHECK WINDOW VERTICES
BPRT.SC DAM.KNDXXX	. GET/CHECK/PROCESS MACRO COMMAND (CALLS KMXXED)
BPRT.SC DAM.KNDZON	. GET/CHECK UTM PROJECTION ZONE
BPRT.SC DAM.KNDODD	. SADD -- DYNAMIC SADD
BPRT.SC DAM.KNDOAS	. SASO -- DYNAMIC SASO
BPRT.SC DAM.KNDQBR	. SBRKPT -- DYNAMIC SBRKPT
BPRT.SC DAM.KNDQFR	. SFREE -- DYNAMIC SFREE
BPRT.SC DAM.KNDQLO	. SLOG -- DYNAMIC SLOG
BPRT.SC DAM.KMXXED	. EDIT ACTUAL SPECS INTO MACRO COMMAND DEFINITION
BPRT.SC DAM.KMXXGS	. GET/EVALUATE ACTUAL SPEC FOR MACRO COMMAND

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDAL1
002

```

C          0 <= SCAN LINE <= 2500
C          0 <= SCAN SAMPLE <= 3500
C          20. <= LATITUDE <= 80.
C          60. <= LONGITUDE <= 180.
C          0 <= EASTING <= 1000000.
C          0 <= NORTHING <= 9000000.

```

3. A MISSING OR EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

C GLOBAL DECLARATIONS
C -----
C

```

INCLUDE KOMXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST     & COMMON ERTS SCENE PARAMETERS
INCLUDE KOMFIT.LIST    & COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE TRFORM.LIST    & DEFINE COORDINATE TRANSFORMATION FUNCTIONS
INCLUDE KOMIHW.LIST    & COMMON INPUT WINDOW PACKETS
INCLUDE KOMOWW.LIST    & COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDEF.LIST    & DEFINE STRUCTURE OF WINDOW PACKETS

```

C LOCAL DECLARATIONS
C -----
C

```

INTEGER NDSAVE          & SAVE AREA FOR CONTENTS OF NDTOTL ON ENTRY
INTEGER ITEMP          & TEMPORARY
INTEGER MSALIN,MSASAM  & MSS ADJUSTED LINE AND SAMPLE NUMBERS
REAL CORLIN,CORSAM    & CORRECTED LINE AND SAMPLE NUMBERS
INTEGER KORDSY        & EARTH COORDINATE SYSTEM ('DEG' / 'KM' / 'MET')
REAL GEDLAT,GEDLON    & GEOGRAPHIC DEGREES LATITUDE & LONGITUDE
REAL STME,STHN        & SCENE TRANSVERSE MERCATOR EAST & NORTH (METERS)
REAL DSTME,DSTHN     & DELTAS USED TO MODIFY OLD TRANSFORM COEFFS
REAL UTME,UTHN        & UNIVERSAL TRANSVERSE MERCATOR EAST & NORTH (MET)

```

C PROCEDURE
C -----
C

CALL TRACE

C GET/CHECK SCANNER (GSFC-ADJUSTED) COORDINATES

```

NDSAVE=NDTOTL
ITEMP='JUNK'
CALL GETSKH(ITEMP,(3), '**NO SPECIFICATIONS --')
IF(ITEMP.NE.'SCA') CALL WARN9(' COORDINATE SYSTEM NOT SCAN --')
CALL GETSIN(MSALIN, 0.2500,**BAD LINE --')
CALL GETSIN(MSASAM, 0.3500,**BAD SAMPLE --')
CORLIN=CORL4A(MSALIN,MSASAM)
CORSAM=CORS4A(MSALIN,MSASAM)

```

C GET/CHECK EARTH COORDINATE SYSTEM

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDALI
003

C

```
KORDSY=' NUL '  
CALL GETSKH(KORDSY,(3),  'NO EARTH COORDINATE SYSTEM --')  
IF(KORDSY.EQ.'DEG') GO TO 400  
IF(KORDSY.EQ.'KM ') GO TO 600  
IF(KORDSY.EQ.'MET') GO TO 610  
CALL WARN5( 'BAD COORDINATE SYSTEM --')  
GO TO 900
```

C

C

C GET/CHECK GEOGRAPHIC COORDINATES (DEGREES)

C

```
400 CALL GETSSX(GEDLAT, 1..20..80..'BAD LATITUDE --')  
CALL GETSSX(GEDLON, 1..60..180..'BAD LONGITUDE --')  
CALL GETSIN(TEMP, +1.-1.'EXTRA ALIGN SPECIFICATION --')  
IF(INDSAVE.NE.NDTOTL) GO TO 900  
CALL ALIGN
```

C

C

C CONFIRM GEOGRAPHIC COORDINATES

C

```
WRITE(6,445) MSALIN,MSASAM,GEDLAT,GEDLON  
445 FORMAT(' ALIGN, SCAN, '.15,'.15,  
& '. DEGREES, '.F9.5,' LAT, '.F9.5,' LON')  
GO TO 900
```

C

C

C GET/CHECK UTM COORDINATES

C

```
600 CF=1E+3      & CONVERT FROM KILOMETRES TO METRES  
GO TO 620  
610 CF=1.        & METRES -- NO CONVERSION NEEDED  
620 IF((UTMCHD.EQ.0.)OR.(ABS(UTMCHD).GT.180.)) CALL MDWARN(  
= 'UTM ZONE NOT DEFINED')  
CALL GETSRL(UTME, CF,0..1E+6..'BAD EASTING --')  
CALL GETSRL(UTMN, CF,0..9E+6..'BAD NORTHING --')  
CALL GETSIN(TEMP, +1.-1.'EXTRA ALIGN SPECIFICATION --')  
CALL G4U(GEDLAT,GEDLON, UTME,UTMN,UTMCHD)  
IF(INDSAVE.NE.NDTOTL) GO TO 900  
CALL ALIGN
```

C

C

C CONFIRM UTM COORDINATES

C

```
WRITE(6,645) MSALIN,MSASAM,UTME,UTMN  
645 FORMAT(' ALIGN, SCAN, '.15,'.15,  
& '. KM, '-3P.F7.3,' EAST, '.F8.3,' NORTH')
```

C

C

C NORMAL RETURN

C

```
900 KORD=' '  
RETURN
```

C

C

C

DAH PACKAGE APPENDIX M
COMMAND ROUTINES

KNDAL I
004

```
C
C
C
C   INTERNAL
C   SUBROUTINE ALIGN
C
C   COMPUTE DELTAS IN STM COORDINATES BETWEEN OLD ALIGNMENT & NEW ALIGNMENT
C
C   IF (ABS(STMCD-GEDLN).GT.4.) STMCD=GEDLN
C   CALL U4G(STM,STMN, GEDLAT,GEDLN,STMCD)
C   DSTME=STM-STM4C(CORLIN,CORSA)
C   DSTMN=STMN-STMN4C(CORLIN,CORSA)
C
C   USE STM DELTAS TO MODIFY TRANSFORMATION COEFFICIENTS
C
C   CORSTM(3)=CORSTM(3)+DSTMN
C   CORSTM(6)=CORSTM(6)+DSTME
C   CALL REVERT(CORSTM,STMCOR)
C
C   MARK ORIGIN AS DESTROYED
C
C   KSYOHN(HORIO)=0      & ORIGINS IN DIFFERENT SYSTEMS NO LONGER CONSISTENT
C   CALL MONOTE('ORIGIN MUST BE RE-ENTERED')
C
C   RECOMPUTE SCENE CENTER
C
C   CALL G4A(CTRLAT,CTRLON,CTRLIN,CTRSAN)
C
C   RETURN
C   END
```


DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KNDATT
002

C -----
C

INCLUDE KOMXQT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST & COMMON ERTS SCENE PARAMETERS
INCLUDE NULCST.LIST & DEFINE NULL CHARACTER STRING

C
C
C LOCAL DECLARATIONS
C -----
C

INTEGER ITEMP & TEMPORARY
INTEGER ISPITD & SIGN OF PITCH ('+' OR '-')
INTEGER INPITD & INTEGER PART OF PITCH
REAL APITD & DECIMAL PART OF PITCH
INTEGER ISROLD & SIGN OF ROLL ('+' OR '-')
INTEGER INROLD & INTEGER PART OF ROLL
REAL AROLD & DECIMAL PART OF ROLL

C
C
C PROCEDURE
C -----
C

CALL TRACE

C
C
C GET/CHECK PITCH & ROLL
C

CALL GETSRL(PITDEG, 1..-2..+2..'BAD PITCH --')
IF(PITDEG.LE.0) GO TO 120
 CALL UNGETS
 CALL GETSKH(ITEMP.(1), NULCST)
 IF(ITEMP.NE.'+') CALL MDWARN('PITCH UNSIGNED, + ASSUMED')
120 CALL GETSRL(ROLDEG, 1..-2..+2..'BAD ROLL --')
IF(ROLDEG.LE.0) GO TO 150
 CALL UNGETS
 CALL GETSKH(ITEMP.(1), NULCST)
 IF(ITEMP.NE.'+') CALL MDWARN('ROLL UNSIGNED, + ASSUMED')
150 CALL GETSIN(ITEMP, +1.-1..'EXTRA ATTITUDE SPECIFICATION --')

C
C
C CONFIRM PITCH & ROLL
C

PITDEG=.01*AIN(100.*PITDEG+SIGN(0.5,PITDEG)) & ROUND TO 2 DECIMAL PLACES
CALL SPLIT(PITDEG,ISPITD,INPITD,APITD)
ROLDEG=.01*AIN(100.*ROLDEG+SIGN(0.5,ROLDEG)) & ROUND TO 2 DECIMAL PLACES
CALL SPLIT(ROLDEG,ISROLD,INROLD,AROLD)
IF(MCFIRM.NE.0) WRITE(6,165)
 & ISPITD,INPITD,APITD,ISROLD,INROLD,AROLD
165 FORMAT(' ATTITUDE, '.A1,11,F3.2,' PITCH, '.A1,11,F3.2,' ROLL')

C
KOND=''
RETURN
END

DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KHDCEN
002

```
C -----
C
C     INTEGER NDSAVE           S SAVE AREA FOR CONTENTS OF NDTOTL ON ENTRY
C     INTEGER KNTEMP          S TEMPORARY
C
C
C PROCEDURE
C -----
C
C     CALL TRACE
C
C
C GET/CHECK SCAN (OSFC-ADJUSTED) COORDINATES
C
C     NDSAVE=NDTOTL
C     KNTEMP=' NUL'
C     CALL GETSKH(KNTEMP,(3), NULCST)
C     IF(KNTEMP.EQ.' NUL') GO TO 400      S NO SPECS -- CONFIRM
C     IF(KNTEMP.NE.'SCA') CALL WARNS( 'COORDINATE SYSTEM NOT SCAN --')
C     IF(KNTEMP.NE.'SCA') GO TO 400
C     CALL GETSRL(CTRLIN, 1..1000..2000..'BAD LINE --')
C     CALL GETSRL(CTRSAM, 1..1000..2000..'BAD SAMPLE --')
C     CALL GETSIN(KNTEMP, >1.-1..'EXTRA CENTER SPECIFICATION --')
C
C
C CONFIRM CENTER COORDINATES
C
C     400 IF(MCFIRM.NE.0) WRITE(6,445) CTRLIN,CTRSAM
C     445 FORMAT(' CENTER. SCAN. ',FB.1,'.',FB.1)
C
C
C NORMAL RETURN
C
C     900 KOMD='
C     RETURN
C     END
```



```

INCLUDE KOMNER.LIST      & COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST     & COMMON CLASSIFICATION INFO
INCLUDE KOMIRT.LIST     & COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
INCLUDE NULCST.LIST     & DEFINE NUL CHARACTER STRING
  
```

```

C
C
C LOCAL DECLARATIONS
C -----
C
  
```

```

      INTEGER KHTYPE      & TRANSFORMATION TYPE SPEC OR CONFIRMATION
      INTEGER NDSAVE     & TEMPORARY SAVE FOR CONTENTS OF NDTOTL ON ENTRY
      INTEGER NLCHAX     & MAXIMUM NUMBER OF LIMIT CHANNELS
      INTEGER NBF        & BUFFER NUMBER
      INTEGER INTENP     & TEMPORARY
  
```

```

C
C
C PROCEDURE
C -----
C
  
```

```

      CALL TRACE
  
```

```

C
C
C ANY SPECIFICATIONS FROM USER?
C
  
```

```

      KHTYPE=' NUL '
      CALL GETSKN(KHTYPE,(1), NULCST)
      IF(KHTYPE.EQ.' NUL ') GO TO 600      & NO SPECS. SO CONFIRM CURRENT ONES
      CALL UNGETS
  
```

```

C
C
C SAVE DIAGNOSTIC COUNT. DELETE OLD TRANSFORM TYPE. CHANNELS & RADIANCE LIMITS
C
  
```

```

      NDSAVE=NDTOTL
      IRRTYP='NUL'
      NLINCH=0
      DO 120 N=1,5
          LIMCH(N)=999      & NULL FLAG
          NBFCHR(N)=999
          LCVLO(N)=9999
          LCVHI(N)=9999
  
```

```

120 CONTINUE
  
```

```

C
C
C GET CHANNEL TRANSFORMATION TYPE
C
  
```

```

      CALL GETSKN(IRRTYP,(3), NULCST)
      IF(IRRTYP.GT.0) GO TO 220      & POSITIVE WORD STARTS WITH ALPHA CHARACTER
  
```

```

C
C
C NO CHANNEL TYPE -- ASSUME 'RAW' (UNTRANSFORMED)
C
  
```

```

      IRRTYP='RAW'
      CALL UNGETS
  
```

```

C
C
C CHECK CHANNEL TYPE & SET MAX ALLOWABLE NUMBER OF EXPLICIT CHANNELS
  
```


DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KHDCNA
004

RETURN
END

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

DETCMA
002

INCLUDE KOMXQT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST & COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST & COMMON CLASSIFICATION INFO
INCLUDE NULCST.LIST & DEFINE NUL CHARACTER STRING

C
C
C LOCAL DECLARATIONS
C -----
C

INTEGER KHTYPE & CHANNEL TYPE SPECIFICATION OR CONFIRMATION
INTEGER NOSAVE & TEMPORARY SAVE FOR CONTENTS OF NOTOTL ON ENTRY
INTEGER NBF & BUFFER NUMBER
INTEGER INTEMP & TEMPORARY

C
C
C PROCEDURE
C -----
C

CALL TRACE

C
C
C ANY SPECIFICATIONS FROM USER?
C

KHTYPE=' NUL '
CALL GETSKH(KHTYPE.(1), NULCST)
IF(KHTYPE.EQ.' NUL ') GO TO 600 & NO SPECS. SO CONFIRM CURRENT ONES
CALL UNGETS

C
C
C SAVE DIAGNOSTIC COUNT. DELETE OLD CHANNEL NUMBERS & RADIANCE LIMITS
C

NOSAVE=NOTOTL
NLIMCH=0
DO 120 N=1.5
 LIMCH(N)=999 & NULL FLAG
 NBFCHR(N)=999
 LCVLO(N)=-9999
 LCVHI(N)=-9999

120 CONTINUE

C
C
C GET/CHECK CHANNEL TYPE -- IF NONE, ASSUME 'DET'
C

CALL GETSKH(KHTYPE.(3), NULCST)
IF(TRUAL(KHTYPE.(1,1))) GO TO 220 & IS FIRST CHAR ALPHA?
 KHTYPE='DET' & NOT ALPHA -- ASSUME CHANNEL TYPE OMITTED
 CALL UNGETS & -- SPEC WAS CHAN * . BACK UP TO GET AS INT
220 IF(KHTYPE.NE.'DET') CALL WARNS(
 ' CHANNEL TYPE NOT DETECTION --')

C
C
C GET/CHECK EXPLICIT CHANNEL NUMBERS
C

DO 360 NBF=1,NERCHA
 CALL GETSIN(LIMCH(NBF), 1,NERCHA,'BAD CHANNEL --')
 IF(LIMCH(NBF).EQ.999) GO TO 360 & NO CHANNEL OR BAD CHANNEL

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

DETCMA
003

```
      NLINCH=NLINCH+1
      IF(NBF.EQ.1) GO TO 360
      DO 340 N=2,NBF
        IF(LINCH(N-1).EQ.LINCH(NBF)) CALL WARN5(
          'DUPLICATE CHANNEL --')
      340 CONTINUE
      360 CONTINUE
      CALL GETSIN(INTERP.  +1.-1.'TOO MANY CHANNELS --')

C
C
C CHECK FOR DIAGNOSTICS
C
      IF(INDTOTL.EQ.NOSAVE) GO TO 600
      NLINCH=0
      LINCH(1)=999
      NBFCHR(1)=999
      GO TO 900

C
C
C CONFIRM CHANNEL TYPE AND CHANNEL NUMBERS
C
      600 IF(MCFIRM.EQ.0) GO TO 900
      WRITE(6,655) (LINCH(N),N=1,NLINCH)
      655 FORMAT(' CHANNEL. DETECT.  .12.T25.12.
        &          T23..  .T29.12.
        &          T27..  .T33.12.
        &          T31..  .T37.12.
        &          T35..  ')

C
C
C NORMAL RETURN
C
      900 KOHD=
      RETURN
      END
```


DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KMDCLE
002

```
C
C
C PROCEDURE
C -----
C
C     CALL TRACE
C
C
C GET/CHECK SPECIFICATION
C
C     ITEMP='WAR'
C     CALL GETSKH(ITEMP.(3), 'NO CLEAR SPEC -- WARNING ASSUMED')
C     IF(ITEMP.EQ.'WAR') GO TO 200
C     IF(ITEMP.EQ.'ERR') GO TO 300
C     CALL WARN5( 'BAD CLEAR SPECIFICATION --')
C     GO TO 90C
C
C
C CLEAR WARNING(S)
C
C 200 CALL MDCLR( NULCST)
C     IF(MCFIRM.NE.0) WRITE(6,215)
C 215 FORMAT(' CLEAR. WARNINGS')
C     GO TO 900
C
C
C CLEAR FATAL ERROR(S)
C
C 300 CALL MDCLRF( NULCST)
C     IF(MCFIRM.NE.0) WRITE(6,315)
C 315 FORMAT(' CLEAR. ERRORS')
C
C
C COMMON RETURN
C
C 900 CALL GETSIN(ITEMP. +1.-1.'EXTRA CLEAR SPECIFICATION --')
C     KOMD=' '
C     RETURN
C     END
```


DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDCOL
003

```

C
C
C INITIALIZE IMPLICIT 'MAXIMUM' COLOR AND NUMBER
C
    CALL NOVCST(MAXKLR.(1).(12),  MINKLR.(1).(12), ' ')
    MAXNUM=MINNUM
C
C
C IS 3RD SPEC FIELD INTEGER?
C
    INTMP=MAXINT
    CALL GETSIN(INTMP,  -9999.+9999.NULCST)  & STILL MAXINT IF NOT INTEGER
    CALL UNGETS
C
C
C IF 3RD SPEC FIELD IS NOT INTEGER, THEN GET OPTIONAL 'MAXIMUM' COLOR FROM IT
C
    IF(INTMP.EQ.MAXINT) CALL GETSKH(MAXKLR.(12),  NULCST)
    MAXIKE=MAXINT
    CALL I4KOLR(MAXIKE,  MAXKLR)  & CONVERT COLOR TO I-K-E
    IF(MAXIKE.EQ.MAXINT) CALL WARN5(  'BAD SECOND COLOR --')
C
C
C GET OPTIONAL 'MAXIMUM' NUMBER (INTEGER) FROM NEXT SPEC FIELD
C
    CALL GETSIN(MAXNUM,  0.ISYMH1.'BAD SECOND NUMBER --')
C
C
C COMPUTE COLORS PER NUMBER
C
    INCIKE=+1
    IF(MINIKE.GT.MAXIKE) INCIKE=-1
    MAXNUM=MIN0(MAXNUM.ISYMH1)
    INCNUM=+1
    IF(MINNUM.GT.MAXNUM) INCNUM=-1
    COINUM=FLOAT(MAXIKE-MINIKE+INCIKE) /  & PREPARE TO INTERPOLATE ...
    & ... OVER INCLUSIVE RANGE
    & FLOAT(MAXNUM-MINNUM+INCNUM)
    IF(ABS(COINUM).GT.1.) CALL MDWARN(
    = 'MORE THAN 1 COLOR PER NUMBER')
C
C
C CHECK FOR EXTRA SPEC FIELDS & FOR DIAGNOSTICS
C
    CALL GETSIN(IN4NUM,  +1.-1.'EXTRA COLOR SPECIFICATION --')
    IF(INDSAVE.NE.NOTOTL) GO TO 900
C
C
C INTERPOLATE I-K-E'S & LOAD INTO COLOR PART (CHAR 6) OF SYMBOL TABLE
C
    AMINIK=FLOAT(MINIKE)+0.0001  & INTERPOLATE UP FROM LOW SIDE OF MINIKE
    IF(MINIKE.GT.MAXIKE)
    & AMINIK=FLOAT(MINIKE)+0.9999  & INTERPOLATE DN FROM HIGH SIDE OF MINIKE
    DO 300 NUM=MINNUM,MAXNUM,INCNUM
        CALL PUTICE(KSYM(NUM+1).(6),  & INTERPOLATE I-K-E
        IFIX(AMINIK*FLOAT(NUM-MINNUM)*COINUM))
    300 CONTINUE

```


DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDCOP
002

C -----

C

CALL TRACE

C

CALL GETSIN(NCOPY. 0.5.'BAD NUMBER OF COPIES (MORE THAN 5) --')
CALL GETSIN(ITEMP. +1.-1.'EXTRA COPY SPECIFICATION --')
IF(NCFIRM.NE.0) WRITE(6,125) NCOPY
125 FORMAT(' COPIES. ',I2)
KOND=' '
RETURN
END

DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KMDCOU
002

CALL TRACE

C
C
C

GET/CHECK COUNT SPECIFICATION

CALL GETSIN(KTIPIX, 0.12, 'BAD COUNT PER PIXEL --')
CALL GETSIN(TEMP, -1.-1, 'EXTRA COUNT SPECIFICATION --')
IF(NCFIRM.NE.0) WRITE(0.120) KTIPIX
120 FORMAT(' COUNT, '.12.' PER PIXEL')
KOND=
RETURN
END

DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KHOCRO
002

```
      DOUBLE PRECISION KOLNAM      & COLOR NAME (12 CHARS)
      INTEGER KFRSUB(10)      & FREQ SUBTOTALS
C
C
C PROCEDURE
C -----
C
C      CALL TRACE
C
C
C CHECK IF KONTBL LOADED WITH FREQUENCY INFO
C
      IF(MCOLOR.NE.0) GO TO 110
      CALL MDWARN(
-      'CROSSTAB COMMAND NOT ALLOWED (COLOR MODE NOT ON)')
      GO TO 900
110 CONTINUE
      IF((KLSTYP.EQ.0).OR.      & NO CLASS TYPE FROM PREVIOUS DISPLAY
& (KTBLTY.NE.'FREQ')) & NO FREQ TABLES LOADED FROM PREVIOUS DISPLAY
&      CALL MDWARN('NO PREVIOUS WINDOW TO CROSSTAB')
C
C
C DRAIN SPECS FOR CURRENT COMMAND
C
      CALL GETSIN(INTMP.  >1.-1.'EXTRA CROSSTAB SPECIFICATION ---')
C
C
C CHECK FOR DIAGNOSTICS
C
      IF(MDATAC.NE.0) GO TO 900      & DATA/CHECKOUT MODE
      IF(INDOTL.NE.0) GO TO 800
C
C
C USE PREVIOUS WINDOW NUMBER FOR PAGE HEADING
C
      NWTEMP=NWINDOW      & SAVE CURRENT WINDOW NUMBER
      NWINDOW=KTBLNW      & WINDOW NUMBER FROM FREQUENCY TABLES
C
C
C PRINT PAGE/WINDOW HEADINGS
C
      WRITE(6,115) NWINDOW.MTERAL
115 FORMAT(' WINDOW NUMBER '.J3.6X.'CROSSTAB'.6X.4A6)
      CALL MOUNIT(4,10)
      WRITE(10,115) NWINDOW.MTERAL
C
C
C RESTORE CURRENT WINDOW NUMBER
C
      NWINDOW=NWTEMP
C
C
C PRINT TABLE HEADINGS
C
      CALL IDERT(8)
      CALL IDERT(10)
```

DAM PACKAGE APPENDIX M
 COMMAND ROUTINES

KMDCRO
 003

```

    IF(MBATCH.EQ.0)
      & WRITE(6,125)
      WRITE(10,125)
    125 FORMAT(
      & 'OCOLOR                                INTENSITY'//
      & '                                10%  20%  30%  40%  50%'
      & '                                60%  70%  80%  90% 100% TOTAL')
      WRITE(6,135)
      WRITE(10,135)
    135 FORMAT(1X)
  C
  C
  C INITIALIZE SUBTOTALS
  C
    DO 340 IIE=0.9
      KFRSUB(IIE+1)=0
    340 CONTINUE
  C
  C
  C PRINT CROSSTABULATIONS
  C
    DO 600 IKE=0.10      & BLUE THROUGH MAGENTA
      CALL KOLR(I(KOLNAM, IKE))
      KFRTOT=0
      DO 400 IIE=0.9      & 10% THROUGH 100%
        KFRTOT=KFRTOT+KFCRO(IIE+1,IKE+1)
        KFRSUB(IIE+1)=KFRSUB(IIE+1)+KFCRO(IIE+1,IKE+1)
      400 CONTINUE
      IF(MBATCH.EQ.0) WRITE(6,425)
      & WRITE(10,425) KOLNAM.(KFCRO(IIE+1,IKE+1),IIE=0.9),KFRTOT
      WRITE(10,425) KOLNAM.(KFCRO(IIE+1,IKE+1),IIE=0.9),KFRTOT
    425 FORMAT(1X,A9,1X,10I6,17)
    600 CONTINUE
  C
  C
  C PRINT TOTALS
  C
    KFRTOT=0
    DO 610 IIE=0.9
      KFRTOT=KFRTOT+KFRSUB(IIE+1)
    610 CONTINUE
    IF(MBATCH.EQ.0)
      & WRITE(6,615) (KFRSUB(IIE+1),IIE=0.9),KFRTOT
      WRITE(10,615) (KFRSUB(IIE+1),IIE=0.9),KFRTOT
    615 FORMAT('OTOTAL',5X,10I6,17)
    GO TO 900
  C
  C
  C CHECK DIAGNOSTIC COUNTERS
  C
    800 IF(INDWARN.EQ.0) GO TO 820
    CALL HDNOTE('PREVIOUS WARNINGS -- NO CROSSTAB GENERATED')
    IF(MBATCH.EQ.0) WRITE(6,815)
    815 FORMAT('...TRY AGAIN')
    GO TO 890
    820 IF(INDFATL.EQ.0) GO TO 850
  
```

DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KHOCRO
004

```
      CALL MONOTE( 'PREVIOUS FATAL ERRORS -- NO CROSSTAB GENERATED')
      GO TO 890
850 IF(MCHECK.EQ.0) GO TO 890
      CALL MONOTE( 'CHECKOUT MODE -- NO CROSSTAB GENERATED')
C
C
C CLEAR WARNINGS
C
C 890 CALL NOCLR( NULCST)
C
C
C RETURN
C
C 900 KOHD=
      RETURN
      END
```

SUBROUTINE KMODEN(3 GET/CHECK DENSITY LIMITS
U KOMD) 3 I: FIRST 3 CHARS OF COMMAND 0: SPACES

C
C
C HISTORY
C -----
C
C E H SCHLOSSER LEC 07/03/73 NUMERIC OPTION
C E H SCHLOSSER LEC 11/29/75 ALPHANUMERIC COMMAND
C E H SCHLOSSER LEC 07/17/78 ADD COUNT SPECIFICATION
C E H SCHLOSSER LEC 03/22/79 CONFIRM COUNT AS SEPARATE COMMAND
C E H SCHLOSSER LEC 11/13/79 DOCUMENT SPECIFICATION RANGES
C
C METHOD
C -----
C
C UPDATE MINIMUM & MAXIMUM DENSITY FROM UNIT 5. IF SPECIFIED. AND CONFIRM.
C
C MACHINE-DEPENDENT CODE
C -----
C
C NONE.
C
C EXTERNAL REFERENCES
C -----
C
C GETSIN 3 GET INTEGER DATA FIELD FROM UNIT 5
C
C EXCEPTIONS
C -----
C
C 1. IF THE CURRENT DETECTION FILE TYPE IS NOT 'DEN'. THEN A WARNING IS
C GENERATED.
C
C 2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
C 'WARNING' DIAGNOSTICS:
C 0 <= MIN DENSITY <= 19
C MIN DENSITY <= MAX DENSITY <= 19
C 0 <= COUNT/PIXEL <= 12
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST 3 COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
C INCLUDE KOMKLS.LIST 3 COMMON CLASSIFICATION INFO
C
C LOCAL DECLARATIONS
C -----
C

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KHDDEN
002

```
      INTEGER INTMP      & TEMPORARY
C
C
C PROCEDURE
C -----
C
C      CALL TRACE
C
C
C DENSITY COMMAND IS VALID ONLY IF DETECTION FILE TYPE IS 'DEN'
C
C      IF(KLSTYP.EQ.'DEN') GO TO 200
C          CALL MDWARN( 'INVALID WITHOUT DENSITY DETECTION FILE ---')
C          GO TO 900
C
C
C GET/CHECK MINIMUM DENSITY
C
C 200 LCVLO1=MAX0(0.LCVLO1)
C     LCVLO1=MIN0(19.LCVLO1)
C     CALL GETSIN(LCVLO1, 0.19,'BAD MINIMUM DENSITY ---')
C
C
C GET/CHECK MAXIMUM DENSITY
C
C     LCVH11=MAX0(0.LCVH11)
C     LCVH11=MIN0(19.LCVH11)
C     CALL GETSIN(LCVH11, 0.19,'BAD MAXIMUM DENSITY ---')
C
C
C GET/CHECK COUNT PER PIXEL
C
C     KTIPIX=MAX0(0.KTIPIX)
C     KTIPIX=MIN0(12.KTIPIX)
C     INTMP=-9999
C     CALL GETSIN(INTMP, 0.12,'BAD COUNT PER PIXEL ---')
C     IF(INTMP.NE.-9999) KTIPIX=INTMP
C     CALL GETSIN(INTMP, +1.-1,'EXTRA SPECIFICATION ---')
C     IF(INTMP.EQ.-9999) GO TO 300      & COUNT SPECIFICATION MISSING?
C
C
C CONFIRM DENSITY LIMITS & COUNT PER PIXEL
C
C     IF(INCFIRM.NE.0) WRITE(6,225) LCVLO1,LCVH11,KTIPIX
C 225     FORMAT(' DENSITY, '.J2,' MINIMUM, '.J2,' MAXIMUM',
C 6         ' COUNT, '.I2,' PER PIXEL')
C     GO TO 900
C
C
C CONFIRM DENSITY LIMITS ONLY
C
C 300 IF(INCFIRM.NE.0) WRITE(6,325) LCVLO1,LCVH11
C 325 FORMAT(' DENSITY, '.J2,' MINIMUM, '.J2,' MAXIMUM')
C
C
```

**DAM PACKAGE APPENDIX H
COMMAND ROUTINES**

**KHDDEN
003**

C DONE

C

**900 KOMD-
RETURN
END**

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMOEXP
002

```
C -----
C
C   INTEGER NAMELT(5)   & NAME OF DISK SYMBOLIC ELEMENT CONTAINING EXPLANATION
C   DATA (NAMELT(N),N=1,2) // '   DAM.EXP-' //
C   INTEGER KHRI       & 1ST CHARACTER OF USER SPECIFICATION
C
C
C PROCEDURE
C -----
C
C   CALL TRACE
C
C
C INITIALIZE 'EXP' SPEC BEFORE TRYING TO GET 1ST SPEC
C
C   NAMELT(3)='EXP '
C
C
C GET NEXT SPECIFICATION
C
C 120 CALL GETSKH(NAMELT(3),(17), NULCST)
C   CALL PUTCHR(NAMELT(3),(18), ' ') & TERMINATE WITH BLANK
C   IF(NAMELT(3).EQ.' NUL') GO TO 900 & NO MORE SPECS
C
C
C IF 1ST CHAR IS 'S' CHANGE IT TO 'S'
C
C   CALL GETCHR(KHRI, NAMELT(3),(1))
C   IF(KHRI.EQ.'S') CALL PUTCHR(NAMELT(3),(1), 'S') & EXEC COMMAND
C
C
C WRITE EXPLANATION IF AVAILABLE
C
C   CALL RITADD(6,NAMELT,0,$180)
C   GO TO 200
C
C
C EXPLANATION HAS NOT AVAILABLE -- TRUNCATE SPEC TO 3 CHARS & TRY AGAIN
C
C 180 CALL MOVCST(NAMELT(3),(4),(13),
C   ' ',(1),(1),' ')
C   CALL RITADD(6,NAMELT,0,$800)
C
C
C INITIALIZE NUL SPEC BEFORE TRYING TO GET NEXT SPEC
C
C 200 NAMELT(3)=' NUL '
C   GO TO 120
C
C
C EXPLANATION STILL NOT AVAILABLE
C
C 800 CALL WARN5( 'BAD EXPLAIN SPECIFICATION ---' )
C   GO TO 200
C
C
```

DAH PACKAGE APPENDIX H
COMMAND ROUTINES

KHDEXP
003

C COMMON RETURN

C

900 KOND= ' .
RETURN
END

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KHDF1
001

SUBROUTINE KHDF1(I) TERMINATE IF .FI BLOCK
U KOND) , I: FIRST 3 CHARS OF COMMAND O: SPACES

```
C
C
C HISTORY
C -----
C      E H SCHLOSSER      LEC      12/05/79      RQHTS/DESIGN/CODE
C
C METHOD
C -----
C      DO NOTHING.
C
C MACHINE-DEPENDENT CODE
C -----
C      NONE.
C
C EXTERNAL REFERENCES
C -----
C      GETSIN      I GET INTEGER DATA FIELD FROM UNIT 5
C
C EXCEPTIONS
C -----
C      1. IF ANY USER SPECIFICATION(S) ARE PRESENT, A 'WARNING' DIAGNOSTIC
C         IS GENERATED.
C
C GLOBAL DECLARATIONS
C -----
C      NONE.
C
C LOCAL DECLARATIONS
C -----
C      INTEGER INTEMP      I TEMPORARY
C
C PROCEDURE
C -----
C
C      CALL TRACE
C
C CHECK FOR EXTRA SPECIFICATION
```

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDF1
002

```
C      CALL GETSIN/INTMP.  *1.-1.*EXTRA FI SPECIFICATION --*)
C
C      NORMAL RETURN
C
      KOND=0
      RETURN
      END
```


DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KNDGEO
002

```
C -----  
C  
C     INTEGER INTMP      & TEMPORARY  
C     INTEGER KNTMP     & TEMPORARY  
C  
C PROCEDURE  
C -----  
C  
C     CALL TRACE  
C  
C ANY SPECIFICATIONS FROM USER?  
C  
C     KNTMP=' NUL '  
C     CALL GETSKM(KNTMP,(3), NULCST)  
C     IF(KMTYPE.EQ.' NUL ') GO TO 800      & NO SPECS. SO CONFIRM CURRENT ONES  
C  
C CHECK GEOMETRY  
C  
C     IF((KNTMP.NE.'ERT').AND.  
C       (KNTMP.NE.'HOM').AND.  
C       (KNTMP.NE.'LCC').AND.  
C       (KNTMP.NE.'PS ').AND.  
C       (KNTMP.NE.'SOM').AND.  
C       (KNTMP.NE.'UTM')) KNTMP='BAD'  
C     IF(KNTMP.EQ.'BAD') CALL WARN9('BAD GEOMETRY ---')  
C     IF(KNTMP.NE.'BAD') NERGEO=KNTMP  
C     CALL GETSIN(INTMP, >1,-1,'EXTRA GEOMETRY SPECIFICATION ---')  
C  
C CONFIRM GEOMETRY  
C  
C     800 IF(MCFICH.NE.0) WRITE(6,855) NERGEO  
C     855 FORMAT(' GEOMETRY, ',A3)  
C  
C     KOND='  
C     RETURN  
C     END
```


DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMOHEA
002

```
C
C
C PROCEDURE
C -----
C
C     CALL TRACE
C
C
C GET/CHECK LINE NUMBER & HEADING TEXT
C
C     NHD1=0
C     CALL GETSIN(NHD1, 1,2,'BAD HEADING LINE NUMBER (NOT 1 OR 2) --')
C     IF(NHD1.NE.0) GO TO 200
C         NHD1=1
C         NHD2=2
C         GO TO 300
C 200 NHD2=NHD1
C     CALL GETSKH(JHDG(1,NHD1),(72), NULCST)
C
C
C CONFIRM HEADING LINE(S)
C
C 300 IF(MCFIRM.EQ.0) GO TO 900
C     DO 400 NHD=NHD1,NHD2
C         WRITE(6,355) NHD,(JHDG(NHD,NHD),NHD=1,12)
C 355     FORMAT(' HEADING, ',11,' ',12A6)
C 400     CONTINUE
C
C
C COMMON RETURN
C
C 900 KOMO=' '
C     RETURN
C     END
```


C
 C LOCAL DECLARATIONS
 C -----
 C

INTEGER KOND	3	CONDITION SPECIFIED BY USER ('OFF' OR 'ON')
INTEGER KONDSW	3	CONDITION SWITCH SETTING (0 OR 1 IF VALID)
INTEGER MODE	3	MODE SPECIFIED BY USER
INTEGER MODESW	3	MODE SWITCH SETTING (0 OR 1 IF VALID)
INTEGER INTMP	3	TEMPORARY
INTEGER NSKIP	3	NUMBER OF COMMANDS SKIPPED
INTEGER KMSKP	3	NAME OF COMMAND BEING SKIPPED

C
 C
 C PROCEDURE
 C -----
 C
 C

CALL TRACE

C
 C
 C GET/CHECK CONDITION 'OFF' OR 'ON' & CONVERT TO CONDITION SWITCH
 C

```

KOND=' NUL '
CALL GETSKM(KOND.(3). NULCST)
KONDSW=MAXINT
IF(KOND.EQ.'OFF') KONDSW=0
IF(KOND.EQ.'ON ') KONDSW=1
IF(KONDSW.EQ.MAXINT) CALL WARN5(
- 'BAD CONDITION (NOT ON OR OFF) --')
  
```

C
 C
 C GET/CHECK MODE & FIND CURRENT VALUE OF ITS MODE SWITCH
 C

```

MODE=' NUL '
CALL GETSKM(MODE.(3). NULCST)
MODESW=MAXINT
IF(MODE.EQ.'BAT') MODESW=MIND(MBATCH.1) 3 BATCH
IF(MODE.EQ.'CHE') MODESW=MIND(MCHECK.1) 3 CHECKOUT
IF(MODE.EQ.'COL') MODESW=MIND(MCOLOR.1) 3 COLOR
IF(MODE.EQ.'CCN') MODESW=MIND(MCFIRM.1) 3 CONFIRM
IF(MODE.EQ.'DUM') MODESW=MIND(MDUMP.1) 3 DUMP
IF(MODE.EQ.'ECH') MODESW=MIND(MECHO.1) 3 ECHO
IF(MODE.EQ.'LEG') MODESW=MIND(MLEGND.1) 3 LEGEND
IF(MODE.EQ.'PRO') MODESW=MIND(MPROMT.1) 3 PROMPT
IF(MODE.EQ.'TRA') MODESW=MIND(MTRACE.1) 3 TRACE
IF(MODESW.EQ.MAXINT) CALL WARN5( 'BAD MODE --')
CALL GET5IN(INTMP. 1.-1.'EXTRA IF SPECIFICATION --')
  
```

C
 C
 C IF SPECIFICATION VALID & KONDSW <> MODESW. THEN SKIP COMMANDS INSIDE BLOCK
 C

```

IF((KONDSW.EQ.MAXINT).OR. 3 CONDITION INVALID
& (MODESW.EQ.MAXINT).OR. 3 MODE INVALID
& (KONDSW.EQ.MODESW)) 3 MODE SWITCH IN SPECIFIED CONDITION
& DO TO 900 3 THEN DON'T SKIP COMMANDS IN BLOCK!!
DO 600 NSKIP=1.21
  
```

DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KMDIF
003

```
CALL READS(LSSTAT, NULCST)
CALL GETSKH(KMOSKP,(3), NULCST)
IF(KMOSKP.EQ.'IF ') CALL MDHARN(
  'NESTED IF NOT ALLOWED')
IF(KMOSKP.EQ.'FI ') GO TO 900
800 CONTINUE
CALL MDHARN(
  'BAD IF...FI BLOCK -- MORE THAN 20 COMMANDS OR MISSING FI')
C
C NORMAL RETURN
C
900 KMD= '
RETURN
END
```


INCLUDE KOSYH.LIST & COMMON SYMBOL TABLE
 INCLUDE NULCST.LIST & DEFINE NULL CHARACTER STRING
 INCLUDE MAXINT.LIST & DEFINE MAXIMUM INTEGER

C
 C
 C LOCAL DECLARATIONS
 C -----
 C

NOTE: THE 'MINIMUM' INTENSITY, I-I-E, & NUMBER DECLARED BELOW
 PERTAIN TO THE FIRST INTENSITY & NUMBER SPECIFIED BY THE USER. THE
 'MAXIMUM' INTENSITY, I-I-E, & NUMBER DECLARED BELOW PERTAIN TO THE
 SECOND INTENSITY & NUMBER SPECIFIED BY THE USER. THE ACTUAL VALUES
 OF THE 'MINIMUM' I-I-E & NUMBER ARE NOT NECESSARILY LESS THAN THE
 VALUES OF THE 'MAXIMUM' I-I-E & NUMBER.

INTEGER NDSAVE & TEMPORARY SAVE FOR CONTENTS OF NDTOTL ON ENTRY
 INTEGER INTMP & TEMPORARY
 INTEGER KHTMP & TEMPORARY
 INTEGER MINPCT,MAXPCT & MINIMUM, MAXIMUM INTENSITY PERCENT
 INTEGER MINIIE,MAXIIE & MINIMUM, MAXIMUM INTEGER-INTENSITY-EQUIVALENT
 REAL AMINI & FLOOR OR CEILING OF MINIIE TO INTERPOLATE UP OR DOWN FROM
 INTEGER INCIIE & INCREMENT DIRECTION MINIIE TO MAXIIE (+1 OR 01)
 INTEGER NUM & NUMBER (RAD/DENSITY/COUNT/CLASS,ETC.) FOR INTENSIT
 INTEGER MINNUM,MAXNUM & MINIMUM, MAXIMUM NUMBER
 INTEGER INCNUM & INCREMENT DIRECTION MINNUM TO MAXNUM (+1 OR -1)
 REAL TNINUM & INTENSITIES PER NUMBER (FOR INTERPOLATION)

C
 C
 C PROCEDURE
 C -----
 C

CALL TRACE

C
 C
 C INITIALIZE
 C

IF(MCOLOR.NE.0) GO TO 120
 CALL HDWARN:
 'INTENSITY COMMAND NOT ALLOWED (COLOR MODE NOT ON)'
 GO TO 900
 120 NDSAVE=NDTOTL

C
 C
 C GET OPTIONAL 'MINIMUM' INTENSITY (I) FROM 1ST SPEC FIELD
 C

KHTE = ' NUL '
 CALL GETSKH(KHTMP,(4), NULCST)
 IF(KHTMP.EQ.' NUL ') GO TO 850 & NO SPECS. SO CONFIRM EVERYTHING
 LOCPCT=LCHREQ(KHTMP,(1),(4),'8')
 KODTYP='ERR'
 IF(LOCPCT.GT.1) CALL DCODE(INTMP,RLTEMP,KODTYP,
 KHTMP,(1),(LOCPCT-1))
 IF(INTMP.LT.1) KODTYP='ERR'
 IF(INTMP.GT.100) KODTYP='ERR'
 IF(KODTYP.EQ.'IN') MINIIE=(INTMP-1)/10
 IF(KODTYP.NE.'IN') CALL WARN5: 'BAD FIRST INTENSITY --'


```

C
C
C GET REQUIRED 'MINIMUM' NUMBER (INTEGER) FROM 2ND SPEC FIELD
C
  MINNUM=MAXINT
  CALL GETSIN(MINNUM, 0, ISYMH1, 'BAD FIRST NUMBER --')
C
C INITIALIZE IMPLICIT 'MAXIMUM' INTENSITY AND NUMBER
C
  MAXIIE=MINIIE
  MAXNUM=MINNUM
C
C IS 3RD SPEC FIELD INTEGER?
C
  INTEMP=MAXINT
  CALL GETSIN(INTEMP, -9999, +9999, NULCST)  & STILL MAXINT IF NOT INTEGER
  CALL UNGET5
C
C IF 3RD SPEC FIELD NOT INTEGER, GET OPTIONAL 'MAX' INTENSITY (S) FROM IT
C
  KHTEMP=' NUL '
  IF (INTEMP.EQ.MAXINT) CALL GETSKH(KHTEMP,(4), NULCST)
  IF (KHTEMP.EQ.' NUL ') GO TO 220
  LOCPCT=LCHREQ(KHTEMP,(1),(4), 'S')
  KODTYP='ERR'
  IF (LOCPCT.GT.1) CALL DCODE(INTEMP, RLTEMP, KODTYP,
    KHTEMP,(1),(LOCPCT-1))
  IF (INTEMP.LT.1) KODTYP='ERR'
  IF (INTEMP.GT.100) KODTYP='ERR'
  IF (KODTYP.EQ.'IN') MAXIIE=(INTEMP-1)/10
  IF (KODTYP.NE.'IN') CALL WARN5('BAD SECOND INTENSITY --')
220 CONTINUE
C
C
C GET OPTIONAL 'MAXIMUM' NUMBER (INTEGER) FROM NEXT SPEC FIELD
C
  CALL GETSIN(MAXNUM, 0, ISYMH1, 'BAD SECOND NUMBER --')
C
C COMPUTE INTENSITIES PER NUMBER
C
  INCIIE=+1
  IF (MINIIE.GT.MAXIIE) INCIIE=-1
  MAXNUM=MIN0(MAXNUM, ISYMH1)
  INCNUM=+1
  IF (MINNUM.GT.MAXNUM) INCNUM=-1
  TNINUM=FLOAT(MAXIIE-MINIIE+INCIIE) / & PREPARE TO INTERPOLATE ...
  & ... OVER INCLUSIVE RANGE
  & FLOAT(MAXNUM-MINNUM+INCNUM)
  IF (ABS(TNINUM).GT.1.) CALL MDHARN(
    'MORE THAN 1 INTENSITY PER NUMBER')
C
C CHECK FOR EXTRA SPEC FIELDS & FOR DIAGNOSTICS

```

DAM PACKAGE APPENDIX M
 COMMAND ROUTINES

KMOINT
 804

```

C      CALL GETSIN(INTERP.  +1.-1.'EXTRA INTENSITY SPECIFICATION --')
      IF(NOSAVE.NE.NDOTOTL) GO TO 900
C
C      INTERPOLATE I-I-E'S & LOAD INTO INTENSITY PART(CHAR 5) OF SYMBOL TABLE
C
      AMINII=FLOAT(MINIIE)+0.0001  & INTERPOLATE UP FROM LOW SIDE OF MINIIE
      IF(MINIIE.GT.MAXIIE)
      & AMINII=FLOAT(MINIIE)+0.9999  & INTERPOLATE DN FROM HIGH SIDE OF MINIIE
      DO 300 NUM=MINNUM.MAXNUM.INCNUM
      CALL PUTICE(KSYM(NUM+1).(5).  & INTERPOLATE I-I-E
      IFIX(AMINII-FLOAT(NUM-MINNUM)*TNINUM))
      300 CONTINUE
C
C      CONFIRM INTENSITIES
C
      MINPCT=(MINIIE+1)*10
      MAXPCT=(MAXIIE+1)*10
      IF(MCFIRM.NE.0)
      & WRITE(6,325) MINPCT.MINNUM.MAXPCT.MAXNUM
      325  FORMAT(' INTENSITY.  '.14.'8'.'.J3.'.  '.14.'8'.'.J3)
      GO TO 900
C
C      NO INTENSITIES SPECIFIED -- CONFIRM ALL INTENSITIES
C
      850 IF(MCFIRM.NE.0) CALL ALLINT
C
C      NORMAL RETURN
C
      900 KOND=' '
      RETURN
C
C
C
C
C      INTERNAL
      SUBROUTINE ALLINT  & CONFIRM ALL CURRENT INTENSITIES
C
      INTEGER NXTIIE  & NEXT I-I-E
C
      MINNUM=0
      CALL GETICE(MINIIE.KSYM(1).(5))
      DO 200 MAXNUM=0.ISYMH1
      CALL GETICE(NXTIIE.  KSYM(MAXNUM+2).(5))
      IF(NXTIIE.EQ.MINIIE) GO TO 200  & NEXT I-I-E SAME AS CURRENT
      MINPCT=(MINIIE+1)*10
      IF(MINNUM.EQ.MAXNUM)
      & WRITE(6,145) MINPCT.MINNUM
      145  FORMAT(' INTENSITY.  '.14.'8'.'.J3)
      IF(MINNUM.NE.MAXNUM)
  
```

DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KNDINT
005

```
      6          WRITE(6,165) MINPCT,MINNUM,MAXNUM  
165          FORMAT(' INTENSITY, '.14,'S','.','.J3,'.','.J3)  
          MINNUM=MAXNUM+1  
          MINIE=NXTIE  
200 CONTINUE  
      RETURN  
C  
C  
      END
```


C GLOBAL DECLARATIONS

C -----
 C

INCLUDE KOKXQT.LIST : COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
 INCLUDE KOKNER.LIST : COMMON ERTS SCENE PARAMETERS
 INCLUDE KOKIRT.LIST : COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
 INCLUDE NULCST.LIST : DEFINE NULL CHARACTER STRING

C
 C LOCAL DECLARATIONS
 C -----
 C

REAL TBIAS(2) : TEMP STORAGE FOR BIASES AS REAL NUMBERS
 INTEGER JFHT(8) : DYNAMIC FORMAT -- JFHT(8) CHANGES
 C (UNIVAC MD) 1111122222333334444455555666667777788888
 DATA JFHT/'(9M LINEAR, .11.9M. WEIGHTS, 000000(2M, .FB.4)1)'/

C
 C PROCEDURE
 C -----
 C

CALL TRACE

C
 C RETRIEVE CURRENT LINEAR BIASES
 C

TBIAS(1)=FLOAT(LRTBI2(1))/2.**12
 TBIAS(2)=FLOAT(RTBI2(2))/2.**12

C
 C GET/CHECK LINEAR TRANSFORMED CHANNEL NUMBER
 C

NL='NONE'
 CALL GETSIN(NL, 1,2,'BAD LINEAR CHANNEL NUMBER ---')
 IF(NL.EQ.'NONE') GO TO 850 : NONE OR INVALID

C
 C GET TRANSFORMATION COEFFICIENT TYPE ('WEIGHTS'/'GAIN'/'BIAS')
 C

300 ITEMP='NONE'
 CALL GETSKH(ITEMP,(3), NULCST)

C
 C GET/CHECK/CONFIRM WEIGHTS
 C

IF(ITEMP.NE.'WEI') GO TO 400
 DO 350 NCH=1,NERCHA
 CALL GETSRL(RTLWGT(NCH,NL),
 1..-10..*10..'BAD WEIGHT ---')
 350 CONTINUE
 CALL CST4IN(JFHT(8),11,(8), NERCHA,(8))
 IF(INCFIRM.NE.0) WRITE(6,JFHT)
 NL,(RTLWGT(NCH,NL),NCH=1,NERCHA)
 GO TO 700

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDLIN
003

```

C GET/CHECK/CONFIRM GAIN
C
400 IF(ITEMP.NE.'GAI') GO TO 500
      CALL GETSRL(RTLOAN(NL), 1.,-10.,+10.,'BAD GAIN --')
      IF(NCFIRM.NE.0) WRITE(6,425) NL,RTLOAN(NL)
425  FORMAT(' LINEAR. '.11.'. GAIN. '.F9.3)
      GO TO 700

C
C
C GET/CHECK/CONFIRM BIAS
C
500 IF(ITEMP.NE.'BIA') GO TO 600
      CALL GETSRL(TBIAS(NL), 1.,-100.,+100.,'BAD BIAS --')
      LRTBIZ(NL)=TBIAS(NL)*2.**12
      IF(NCFIRM.NE.0) WRITE(6,525) NL,TBIAS(NL)
525  FORMAT(' LINEAR. '.11.'. BIAS. '.F9.3)
      GO TO 300

C
C
C UPDATE WEIGHTED GAINS
C
700 DO 750 NCH=1,NERCHA
      DO 730 NL=1,2
          LRTWIZ(NCH,NL)=RTLWOT(NCH,NL)*RTLOAN(NL)**2.**12
730  CONTINUE
750 CONTINUE
      GO TO 300

C
C
C FLAG BAD SPECIFICATION
C
800 IF(ITEMP.EQ.'NONE') GO TO 900
      CALL WARN9(' BAD LINEAR SPECIFICATION --')

C
C
C CONFIRM WEIGHTS/GAINS/BIASES
C
850 IF(NCFIRM.EQ.0) GO TO 900
      DO 860 NL=1,2
          CALL CST4IN(JFMT(6),(1),(6), NERCHA,(6))
          WRITE(6,JFMT) NL,(RTLWOT(NCH,NL),NCH=1,NERCHA)
          WRITE(6,425) NL,RTLOAN(NL)
          WRITE(6,525) NL,TBIAS(NL)
860  CONTINUE

C
C
C NORMAL RETURN
C
900 KMD= '
      RETURN
      END

```


DAH PACKAGE APPENDIX M
COMMAND ROUTINES

KMOHER
002

C
C
C PROCEDURE
C -----
C

CALL TRACE
C
IF(UTMCHD.EQ.0.) UTMCHD=36000. & INVALID
OLOCMD=UTMCHD
CALL GETSSX(UTMCHD. 1..90..180..'BAD MERIDIAN --')
IF(MCFIRM.NE.0) WRITE(6,125) UTMCHD
125 FORMAT(' MERIDIAN. .F9.4.' DEGREES CENTER OF TM PROJECTION')
CALL GETSIN(INTERP. +1.-1..'EXTRA MERIDIAN SPECIFICATION --')
IF(OLOCMD.NE.UTMCHD) KSYOWM(WORIG)='NUL' & IF CMD CHANGED MARK ORIG ABSENT
900 KOND=''
RETURN
END

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KHONAM
002

```
C
C
C PROCEDURE
C -----
C
C      CALL TRACE
C
C
C      CALL GETSKH(MTERAL,(24), NULCST)
C      CALL GETSIN(INTERP.  +1,-1,'EXTRA NAME SPECIFICATION --')
C      IF(MCFIRM.NE.0) WRITE(8,125) (MTERAL(N),N=1,4)
125  FORMAT(' NAME. ',4A8)
C      KOHD=
C      RETURN
C      END
```


DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMONEW
002

```
C
C
C PROCEDURE
C -----
C
C     CALL TRACE
C
C
C INITIALIZE 'DAM' SPEC BEFORE TRYING TO GET 1ST SPEC
C
C     NAMELT(3)='DAM'
C
C
C GET NEXT SPECIFICATION
C
C 100 CALL GETSKH(NAMELT(3),(17), NULCST)
C     CALL PUTCHR(NAMELT(3),(18), ' ') & TERMINATE WITH BLANK
C     IF(NAMELT(3).EQ.' NUL') GO TO 900 & NO MORE SPECS
C
C
C WRITE NEWS IF AVAILABLE
C
C     CALL RITADD(6,NAMELT,0,$800)
C
C
C INITIALIZE NUL SPEC BEFORE TRYING TO GET NEXT SPEC
C
C 200 NAMELT(3)=' NUL'
C     GO TO 100
C
C
C NEWS NOT AVAILABLE
C
C 800 CALL WARNS( 'BAD PROGRAM NAME ---')
C     GO TO 200
C
C
C COMMON RETURN
C
C 900 KOMD='
C     RETURN
C     END
```


DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KRDNEX
002

C PROCEDURE

C -----

C

C

C

CALL TRACE

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

KOND=' NUL'
CALL GETSKH(KOND.(3), NULCST)
KONOSW=-999
IF(KOND.EQ.'OFF') KONDSW=0
IF(KOND.EQ.'ON ') KONDSW=1
IF(KONDSW.EQ.-999) CALL WARN5(
= 'BAD CONDITION (NOT ON OR OFF) ---')

GET/CHECK MODE & FIND CURRENT VALUE OF ITS MODE SWITCH

MODE=' NUL'
CALL GETSKH(MODE.(3), NULCST)
MODESW=-999
IF(MODE.EQ.'BAT') MODESW=MIND(MBATCH.1) @ BATCH
IF(MODE.EQ.'CHE') MODESW=MIND(MCHECK.1) @ CHECKOUT
IF(MODE.EQ.'COL') MODESW=MIND(MCOLOR.1) @ COLOR
IF(MODE.EQ.'CON') MODESW=MIND(MCFIRM.1) @ CONFIRM
IF(MODE.EQ.'DUM') MODESW=MIND(MDUMP.1) @ DUMP
IF(MODE.EQ.'ECH') MODESW=MIND(MECHO.1) @ ECHO
IF(MODE.EQ.'LEG') MODESW=MIND(MLEGND.1) @ LEGEND
IF(MODE.EQ.'PRO') MODESW=MIND(MPROMT.1) @ PROMPT
IF(MODE.EQ.'TRA') MODESW=MIND(MTRACE.1) @ TRACE
IF(MODESW.EQ.-999) CALL WARN5('BAD MODE ---')
CALL GETSIN(ITEMP. +1.-1,'EXTRA SPECIFICATION ---')

SKIP NEXT COMMAND IF SPECIFICATION VALID & KONDSW NOT EQUAL MODESW

IF((KONDSW.GT.-999).AND. @ CONDITION VALID
& (MODESW.GT.-999).AND. @ MODE VALID
& (KONDSW.NE.MODESW)) @ MODE SWITCH NOT IN SPECIFIED CONDITION
& CALL READ5(LSSTAT, NULCST) @ THEN SKIP NEXT COMMAND

C NORMAL RETURN

KOND='
RETURN
END

ORIGINAL PAGE IS
OF POOR QUALITY

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KHDOFF
002

```

      INTEGER KHTEMP      & TEMPORARY
C
C
C PROCEDURE
C -----
C
      CALL TRACE
C
C GET FIRST SPECIFICATION
C
      KHTEMP=' NUL'
      CALL GETSKH(KHTEMP,(3), NULCST)
      IF(KHTEMP.EQ.'DEF') GO TO 400
C
C SET SPECIFIED SWITCHES TO OFF (0)
C
      120 IF(KHTEMP.NE.' NUL') CALL UNGETS      & SO SETMOD CAN GET FIRST SPEC
          CALL SETMOD( 0)
          GO TO 900
C
C
C TERMINATE PROCESSING OF DEFAULT COMMANDS
C
      400 IF(NHNDOW.NE.0) GO TO 120      & DEFAULT CAN ONLY BE TURNED OFF ONCE!!!
          NHNDOW=-1
          CALL GETSIN(KHTEMP, +1,-1,'SPEC INVALID AFTER DEFAULT --')
          IF(NDHARN.NE.0) CALL HDFATL( 'INVALID DEFAULT COMMANDS')
          NCARD=0      & READY TO READ FIRST CARD OF USER COMMANDS
          CALL HDUNIT( 0,8)
          WRITE(6,425)
          425 FORMAT(4X,'**USER COMMANDS:**')
C
C
C DONE
C
      900 KOMB='
          RETURN
          END
```


DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDON
002

```
      INTEGER KHTEMP      & TEMPORARY
C
C
C PROCEDURE
C -----
C
C      CALL TRACE
C
C
C GET FIRST SPECIFICATION
C
C      KHTEMP=' NUL'
C      CALL GETSKH(KHTEMP,(3), NULCST)
C      IF(KHTEMP.EQ.'DEF') GO TO 400
C
C
C SET SPECIFIED SWITCHES TO ON (1)
C
C      120 IF(KHTEMP.NE.' NUL') CALL UNGETS
C          CALL SETMOD( 11)
C          GO TO 900
C
C
C INITIATE PROCESSING OF DEFAULT COMMANDS
C
C      400 IF(NMNDOW.NE.0) GO TO 120      & DEFAULT CAN ONLY BE TURNED ON ONCE!!!
C          IF(INCARD.GE.0) CALL MDFATL(
C              - 'DEFAULT COMMANDS MAY NOT FOLLOW USER COMMANDS')
C          WRITE(6,425)
C      425 FORMAT(/'0  **CURRENT DEFAULT COMMANDS: '/')
C          CALL MDCLRW( NULCST)      & CLEAR PREVIOUS WARNINGS
C          CALL GETSIN(KHTEMP, +1,-1,'SPEC INVALID AFTER DEFAULT --')
C
C
C
C DONE
C
C      900 KMD=' '
C          RETURN
C          END
```


DAH PACKAGE APPENDIX M
COMMAND ROUTINES

KNDORI
002

```

C          -200 <= LINE      <= +3000
C          -200 <= SAMPLE   <= +4000
C          CTRLAT-S. <= LATITUDE <= CTRLAT+S.
C          CTRLON-S. <= LONGITUDE <= CTRLON+S.
C          0. <= EASTING   <= 1E+6
C          0. <= NORTHING  <= 9E+6
C
C          3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.
C
C GLOBAL DECLARATIONS
C -----
C
C          INCLUDE KOMXQT.LIST      & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C          INCLUDE KOMNER.LIST      & COMMON ERTS SCENE PARAMETERS
C          INCLUDE KOMFIT.LIST      & COMMON ADJUSTMENT/REGISTRATION PARAMETERS
C          INCLUDE KOMIHW.LIST      & COMMON INPUT WINDOW PACKETS
C          INCLUDE KOMOHM.LIST      & COMMON OUTPUT WINDOW PACKETS
C          INCLUDE WINDEF.LIST      & DEFINE STRUCTURE OF WINDOW PACKETS
C          INCLUDE NULCST.LIST      & DEFINE NULL CHARACTER STRING
C
C LOCAL DECLARATIONS
C -----
C
C          INTEGER INTMP(6)        & TEMPORARY
C          REAL   RLTEMP(2)        & TEMPORARY
C          INTEGER KORDSY          & COORDINATE SYSTEM: 'SCA'/'DEG'/'KM'/'MET'/'PRI'
C
C PROCEDURE
C -----
C
C          CALL TRACE
C
C GET/CHECK COORDINATE SYSTEM
C
C          KORDSY=KSYOWH(WORIO)      & USE OLD COORDINATE SYS IF NONE SPECIFIED
C          CALL GETSKH(KORDSY,(3), NULCST)
C          IF(KORDSY.EQ.'SCA') GO TO 300
C          IF(KORDSY.EQ.'DEG') GO TO 400
C          IF(KORDSY.EQ.'KM ') GO TO 600
C          IF(KORDSY.EQ.'MET') GO TO 610
C          CALL WARN5('BAD COORDINATE SYSTEM --')
C          IF(KORDSY.EQ.'PRI') GO TO 700      & INVALID BUT TRANSFORM ANYWAY!!!
C          GO TO 900
C
C SCANNER (OSFC-ADJUSTED) COORDINATES
C
C          300 CALL GETSIN(MSAOHM(WLIN,WORIO), -200,+3000,'BAD LINE --')
C              CALL GETSIN(MSAOHM(WSAM,WORIO), -200,+4000,'BAD SAMPLE --')
C          320 CALL GVA(DEDONH(WLAT,WORIO),DEDONH(WLON,WORIO),
C              &          FLOAT(MSAOHM(WLIN,WORIO)),
C              &          FLOAT(MSAOHM(WSAM,WORIO)))

```

```

IF(MCFIRM.EQ.0) GO TO 800
WRITE(8,345) MSAOWH(WLIN,WOR10),MSAOWH(WSAM,WOR10)
345 FORMAT(' ORIGIN, SCAN, ',14,' LINE, ',14,' SAMPLE')
WRITE(8,445) GEDOWH(WLAT,WOR10),GEDOWH(WLON,WOR10)
IF((UTMCHD.EQ.0).OR.(ABS(UTMCHD).GT.100.)) GO TO 800
CALL U40(UTMOWH(WEA,WOR10),UTMOWH(WNO,WOR10),
      GEDOWH(WLAT,WOR10),GEDOWH(WLON,WOR10),UTMCHD)
WRITE(8,845) UTMOWH(WEA,WOR10),UTMOWH(WNO,WOR10)
GO TO 800
  
```

C
 C
 C
 C

GEOGRAPHIC COORDINATES (DEGREES)

```

400 CALL GETSSX(GEDOWH(WLAT,WOR10),
      1.,CTRLAT-5.,CTRLAT+5.,'BAD LATITUDE --')
CALL GETSSX(GEDOWH(WLON,WOR10),
      1.,CTRLON-5.,CTRLON+5.,'BAD LONGITUDE --')
CALL A40(RLTEMP(1),RLTEMP(2),
      GEDOWH(WLAT,WOR10),GEDOWH(WLON,WOR10),
      MSAOWH(WLIN,WOR10)+RLTEMP(1)+.5      & ROUND IT!
      MSAOWH(WSAM,WOR10)+RLTEMP(2)+.5
      CALL U40(UTMOWH(WEA,WOR10),UTMOWH(WNO,WOR10),
      GEDOWH(WLAT,WOR10),GEDOWH(WLON,WOR10),UTMCHD)
IF(MCFIRM.EQ.0) GO TO 800
WRITE(8,445) GEDOWH(WLAT,WOR10),GEDOWH(WLON,WOR10)
445 FORMAT(' ORIGIN, DEGREEZ, ',F10.5,' LATITUDE, ',F10.5,' LONGITUDE')
CALL RL21SX(GEDOWH(WLAT,WOR10)+.00001,INTMP(1),3,RLTEMP(1))
CALL RL21SX(GEDOWH(WLON,WOR10)+.00001,INTMP(4),3,RLTEMP(1))
WRITE(8,455) INTMP
455 FORMAT(' ORIGIN, DEGREEZ, ',
      6,14,' ',J2,' ',J2,' LATITUDE, ',
      6,14,' ',J2,' ',J2,' LONGITUDE')
IF((UTMCHD.EQ.0).OR.(ABS(UTMCHD).GT.100.)) GO TO 470
WRITE(8,845) UTMOWH(WEA,WOR10),UTMOWH(WNO,WOR10)
470 WRITE(8,345) MSAOWH(WLIN,WOR10),MSAOWH(WSAM,WOR10)
GO TO 800
  
```

C
 C
 C
 C

UTM COORDINATES

```

800 CF=1E+3      & CONVERT FROM KILOMETRES TO METRES
GO TO 820
810 CF=1.        & METRES -- NO CONVERSION NEEDED
820 IF((UTMCHD.NE.0.).AND.(ABS(UTMCHD).LE.100.)) GO TO 830
CALL MDWARN(' NO UTM ZONE DEFINED')
GO TO 800
830 CALL GETSRL(UTMOWH(WEA,WOR10), CF,0.,1E+6,'BAD EASTING --')
CALL GETSRL(UTMOWH(WNO,WOR10), CF,0.,1E+6,'BAD NORTHING --')
CALL G4U(GEDOWH(WLAT,WOR10),GEDOWH(WLON,WOR10),
      UTMOWH(WEA,WOR10),UTMOWH(WNO,WOR10),UTMCHD)
CALL A40(RLTEMP(1),RLTEMP(2),
      GEDOWH(WLAT,WOR10),GEDOWH(WLON,WOR10),
      MSAOWH(WLIN,WOR10)+RLTEMP(1)+.5      & ROUND IT!
      MSAOWH(WSAM,WOR10)+RLTEMP(2)+.5
      IF(MCFIRM.EQ.0) GO TO 800
WRITE(8,845) UTMOWH(WEA,WOR10),UTMOWH(WNO,WOR10)
  
```

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDORI
004

```
645 FORMAT(' ORIGIN. KM. ', -3P.F7.3, ' EAST. ', F8.3, ' NORTH')
WRITE(6,345) MSAOWH(WLIN,WORIG),MSAOWH(WSAM,WORIG)
GO TO 800

C
C
C PRINT/PLOT DEVICE COORDINATES (FOR DEBUGGING ONLY!!!)
C
700 CALL GET5RL(PPDOWN(WLIN,WORIG), 1..1..6000..'BAD LINE --')
CALL GET5RL(PPDOWN(WCOL,WORIG), 1..1..6000..'BAD COLUMN --')
IF(IRFD.LE.0) GO TO 900      & SCALE NOT CALIBRATED -- CAN'T TRANSFORM PPD
CALL A4P(ADJLIN,ADJSAM,
-      PPDOWN(WLIN,WORIG)+.5,PPDOWN(WCOL,WORIG)+.5)
MSAOWH(WLIN,WORIG)=ADJLIN
MSAOWH(WSAM,WORIG)=ADJSAM
KORDSY='SCA'      & USE SCANNER COORDINATES
GO TO 320

C
C
C STORE COORDINATE SYSTEM
C
800 KSYOWH(WORIG)=KORDSY
CALL GET5IN(INTERP(1), +1.-1..'EXTRA ORIGIN SPECIFICATION --')

C
C
C NORMAL RETURN
C
903 KOMD=' '
RETURN
END
```


DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDPAG
002

C

```
CALL TRACE  
CALL MOUNT( 0.8)  
KHTEMP(1)=' NUL'  
CALL GETSKH(KHTEMP,(48), NULCST)  
IF(KHTEMP(1).NE.' NUL') WRITE(6,145) KHTEMP  
145 FORMAT(6X,8A6/)  
CALL GETSIN(KHTEMP, +1,-1,'EXTRA PAGE SPECIFICATION ---')  
KOND=' '  
RETURN  
END
```



```

INCLUDE KOMLOG.LIST      @ COMMON LOG FILE BUFFER, I/O PKT. POINTERS
COMMON/KOMLU2/KOMLU2(1) @ COMMON POINTERS/FLAGS FOR UNIT 2
COMMON/KOMLU3/KOMLU3(1) @ COMMON POINTERS/FLAGS FOR UNIT 3
COMMON/KOMLU5/KOMLU5(1) @ COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
COMMON/KOML2N/KOML2N(1) @ COMMON I/O PACKETS FOR DETECTION FILES (21-24)
COMMON/KOMNER/KOMNER(1) @ COMMON ERTS SCENE PARAMETERS
COMMON/KOMNET/KOMNET(1) @ COMMON CONTROL NETWORK COORDINATES
COMMON/KOMOWW/KOMOWW(1) @ COMMON OUTPUT WINDOW PACKETS
COMMON/KOMSLM/KOMSLM(1) @ COMMON SPECTRAL LIMITS
COMMON/KOMSYM/KOMSYM(1) @ COMMON SYMBOL TABLE
COMMON/KOMTBL/KOMTBL(1) @ COMMON MULTI-PURPOSE TABLE
INCLUDE KOMXQT.LIST     @ COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST     @ DEFINE NULL CHARACTER STRING
  
```

```

C
C
C LOCAL DECLARATIONS
C -----
C
  
```

```

INTEGER NAMCOM      @ NAME OF COMMON TO 'PEEK' AT
INTEGER LOCPRT      @ LOCATION OF VARIABLE IN COMMON TO PRINT
INTEGER LOCBEG      @ LOCATION OF BEGINNING VARIABLE TO PRINT
INTEGER LOCEND      @ LOCATION OF ENDING VARIABLE TO PRINT
INTEGER LOCINC      @ LOCATION INCREMENT FOR PRINTING
REAL ZERO/.0/      @ ZERO!
LOGICAL PEEKED      @ TRUE IF PEEK CALLED, FALSE IF NOT
  
```

```

C
C
C PROCEDURE
C -----
C
  
```

CALL TRACE

```

C
C
C GET NAME OF COMMON
C
  
```

```

CALL MDCLRW( NULCST)
NAMCOM=' ??? '
CALL GETSKH(NAMCOM,(6), NULCST)
  
```

```

C
C
C GET/CHECK LOCATION OF FIRST VARIABLE TO 'PEEK' AT
C
  
```

```

LOCBEQ=1
CALL GETSIN(LOCBEQ, 1,2500,'BAD PEEK LOCATION --')
  
```

```

C
C
C GET/CHECK LOCATION OF LAST VARIABLE TO 'PEEK' AT
C
  
```

```

LOCEND=LOCBEQ @ DEFAULT IF NOT SPECIFIED
CALL GETSIN(LOCEND, LOCBEQ,LOCBEQ+250,'BAD PEEK END --')
  
```

```

C
C
C GET/CHECK INCREMENT
C
  
```

```

LOCINC=1 @ DEFAULT IF NOT SPECIFIED
  
```

```

    CALL GETSIN(LOCINC, +1,+50,'BAD PEEK INCREMENT --')
    CALL GETSIN(LOCINC, +1,-1,'EXTRA PEEK SPECIFICATION --')
  C
  C
  C GIVE UP IF ANY WARNINGS OCCURRED
  C
    IF(NDWARN.NE.0) GO TO 900
  C
  C
  C FIND REQUESTED COMMON & PRINT
  C
    PEEKED=.FALSE.
    IF(NAMCOM.EQ.'KOMALT') CALL PEEK(KOMALT,KOMALT)
    IF(NAMCOM.EQ.'KOMDET') CALL PEEK(KOMDET,KOMDET)
    IF(NAMCOM.EQ.'KOMFIT') CALL PEEK(KOMFIT,KOMFIT)
    IF(NAMCOM.EQ.'KOMIRT') CALL PEEK(KOMIRT,KOMIRT)
    IF(NAMCOM.EQ.'KOMIHW') CALL PEEK(KOMIHW,KOMIHW)
    IF(NAMCOM.EQ.'KOMKLS') CALL PEEK(KOMKLS,KOMKLS)
    IF(NAMCOM.EQ.'KOMLOG') CALL PEEK(KOMLOG,KOMLOG)
    IF(NAMCOM.EQ.'KOMLU2') CALL PEEK(KOMLU2,KOMLU2)
    IF(NAMCOM.EQ.'KOMLU3') CALL PEEK(KOMLU3,KOMLU3)
    IF(NAMCOM.EQ.'KOMLU5') CALL PEEK(KOMLU5,KOMLU5)
    IF(NAMCOM.EQ.'KOML2N') CALL PEEK(KOML2N,KOML2N)
    IF(NAMCOM.EQ.'KOMNER') CALL PEEK(KOMNER,KOMNER)
    IF(NAMCOM.EQ.'KOMNET') CALL PEEK(KOMNET,KOMNET)
    IF(NAMCOM.EQ.'KOMOHW') CALL PEEK(KOMOHW,KOMOHW)
    IF(NAMCOM.EQ.'KOMSLM') CALL PEEK(KOMSLM,KOMSLM)
    IF(NAMCOM.EQ.'KOMSYH') CALL PEEK(KOMSYH,KOMSYH)
    IF(NAMCOM.EQ.'KOMTBL') CALL PEEK(KOMTBL,KOMTBL)
    IF(NAMCOM.EQ.'KOMXQT') CALL PEEK(KOMXQT,KOMXQT)
    IF(.NOT.PEEKED) CALL MDHARN(
      * 'BAD PEEK COMMON NAME -- --' , CBS4CS(NAMCOM,(1),(6))
  C
  C
  C DCNE
  C
    900 KOMD='
    CALL MDCLRW( NULCST)
    RETURN
  C
  C
  C
  C
  C INTERNAL
  C
  C SUBROUTINE PEEK( 3 PRINT VALUES OF COMMON LOCATIONS
  C I KOM, 3 LABELLED COMMON (INTEGER OR CHARACTER)
  C I COM) 3 LABELLED COMMON (REAL)
  C
  C
  C INTEGER KOM(I) 3 ARGUMENT
  C REAL COM(I) 3 ARGUMENT
  C REAL RLTEMP 3 TEMPORARY
  C INTEGER NFORMAT 3 TYPE OF 'F' FORMAT TO PRINT RLTEMP WITH
  C
    PEEKED=.TRUE.
    DO 300 LOCPRT=LOCBE0,LOCEND,LOCINC
      RLTEMP=COM(LOCPRT)+ZERO 3 FORCE NORMALIZATION IF NOT REAL
  
```

DAH PACKAGE APPENDIX M
COMMAND ROUTINES

KMDPEE
004

```
      NFORMT=1
      IF(ABS(RLTEMP).LT.1.0E+6) NFORMT=2
      IF(ABS(RLTEMP).LT.1.0 ) NFORMT=3
      IF(NFORMT.EQ.1) WRITE(6,
161      161) NAMCON,LOCprt,KOM(LOCprt),RLTEMP,KOM(LOCprt)
      FORMAT(1X,A6,'(',J4,')',.110,1X,F14.0,2X,A6)
      IF(NFORMT.EQ.2) WRITE(6,
162      162) NAMCON,LOCprt,KOM(LOCprt),RLTEMP,KOM(LOCprt)
      FORMAT(1X,A6,'(',J4,')',.110,1X,F14.6,2X,A6)
      IF(NFORMT.EQ.3) WRITE(6,
163      163) NAMCON,LOCprt,KOM(LOCprt),RLTEMP,KOM(LOCprt)
      FORMAT(1X,A6,'(',J4,')',.110,1X,F14.12,2X,A6)
300 CONTINUE
      RETURN
      END
```

SUBROUTINE KMDPLO: 8 GET/CHECK PLOTTER SPECIFICATIONS
U KOMD1 8 1: FIRST 3 CHARS OF COMMAND 0: SPACES

C
C
C HISTORY
C -----
C R E NARVESON LEC 12/04/78 MODIFIED FROM KMDPRI
C E M SCHLOSSER LEC 01/11/79 REWRITE & DOCUMENT
C
C
C METHOD
C -----
C 7???
C
C MACHINE-DEPENDENT CODE
C -----
C FORMAT STATEMENTS ASSUME 8 CHARACTERS PER WORD.
C
C
C EXTERNAL REFERENCES
C -----
C GETSKH 8 GET/CHECK CHARACTER STRING FIELD
C GETSRL 8 GET/CHECK REAL FIELD
C GETSIN 8 GET/CHECK INTEGER FIELD
C WARNS 8 OUTPUT WARNING DIAGNOSTIC FOR PREVIOUS FIELD
C
C
C EXCEPTIONS
C -----
C 1. INVALID FUNCTION NAMES ARE FLAGGED.
C 2. SPECIFICATIONS FOR ALL VALID FUNCTIONS EXCEPT 'DEVICE1' ARE CHECKED.
C 3. NO OTHER CHECKS ARE MADE.
C
C
C GLOBAL DECLARATIONS
C -----
C
C INCLUDE KOMXQT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C INCLUDE KOMPLT.LIST 8 COMMON PLOT FILE/DEVICE CHARACTERISTICS
C INCLUDE NULCST.LIST 8 DEFINE NULL CHARACTER STRING
C
C DIMENSION KOMPLT(S) 8 ??????
C
C 5 WORD COMMON KOMPLT?
C CONTENTS
C UNITS
C INCHES/LENGTH REAL??

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDPLO
002

```

C          INCHES/WIDTH      REAL??
C          DEVICE            CHARACTER
C          'CAL' - CALCOMP
C          'GER' - GERBER
C          MODEL             INTEGER
CC         NUMBER OF PENS    INTEGER
C
C
C LOCAL DECLARATIONS
C -----
C
C          INTEGER NAMFUN      & PLOTTER FUNCTION NAME (SHEET).DEVICE).MODEL).PEN(S))
C          PARAMETER MINPEN = 1      & MINIMUM NO. OF PENS
C          PARAMETER MAXPEN = 3      & MAXIMUM NO. OF PENS
C          PARAMETER XMINLO = 5.0    & MINIMUM INCHES/LENGTH
C          PARAMETER XMAXLO = 99.0   & MAXIMUM INCHES/LENGTH
C          PARAMETER XMINWH = 5.0    & MINIMUM INCHES/WIDTH
C          PARAMETER XMAXWH = 99.0   & MAXIMUM INCHES/WIDTH
C
C
C
C
C PROCEDURE
C -----
C
C          CALL TRACE
C
C
C CHECK IF COMMAND IS NOW LEGAL
C
C          IF(NWINDOW.LT.+1) GO TO 100
C          CALL MDWARN(
C          = 'PLOTTER COMMAND CANNOT BE USED AFTER FIRST WINDOW IS PROCESSED')
C          GO TO 900
C
C
C GET/CHECK PLOTTER FUNCTION TO BE SPECIFIED
C
C 100 NAMFUN='NONE'
C          CALL GETSKH(NAMFUN.(3), NULCST)      & GET 1ST 3 CHARACTERS OF FUNCT NAME
C          IF(NAMFUN.EQ.'SHE') GO TO 300      & SHEET
C          IF(NAMFUN.EQ.'DEV') GO TO 400      & DEVICE
C          IF(NAMFUN.EQ.'MOD') GO TO 500      & MODEL
C          IF(NAMFUN.EQ.'PEN') GO TO 600      & PENS
C          IF(NAMFUN.EQ.'NONE') GO TO 300
C
C
C FLAG BAD FUNCTION & DRAIN SPECS
C
C          CALL WARN( 'BAD PLOTTER FUNCTION --')
C 150 NAMFUN='NONE'
C          CALL GETSKH(NAMFUN.(3), NULCST)
C          IF(NAMFUN.NE.'NONE') GO TO 150
C
C
C

```

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDPLO
003

C GET/CHECK/CONFIRM SHEET UNITS/LENGTH/WIDTH

C

```
300 CALL GETSRL(KOMPLT(1),  
  = 1.0,XMINLG,XMAXLG,'BAD LENGTH ---')  
  CALL GETSRL(KOMPLT(2),  
  = 1.0,XMINWH,XMAXWH,'BAD WIDTH ---')  
  CALL GETSIN(NAMFUN, +1,-1,'EXTRA PLOTTER SPEC ---')  
  IF(MCFIRM.NE.0) WRITE(6,315) KOMPLT(0),KOMPLT(1),KOMPLT(2)  
315 FORMAT(' PLOTTER. SHEET. '.A3,'. '.13,' LONG. '.13,' WIDE')  
  IF(NAMFUN.NE.'NONE') GO TO 900
```

C

C

C GET/CHECK/CONFIRM DEVICE TYPE MNEMONIC

C

```
400 CALL GETSKH(MNEMON,(6), NULCST)  
  IF(MNEMON.EQ.' ') CALL WARNB('BAD DEVICE SPECIFICATION ---')  
  CALL GETSIN(NAMFUN, +1,-1,'EXTRA PLOTTER SPEC ---')  
  IF(MCFIRM.NE.0) WRITE(6,415) MNEMON  
415 FORMAT(' PLOTTER. DEVICE. '.A6)  
  IF(NAMFUN.NE.'NONE') GO TO 900
```

C

C

C MODEL SPECIFICATION

C

```
500 WRITE(6,515)  
515 FORMAT(5X,'MODEL MNEMONICS ARE NOT DEFINED.')
```

```
GO TO 900
```

C

C

C GET/CHECK/CONFIRM NUMBER OF PENS

C

```
600 MALTH=MIND,MALTH,MSALTH) 3 INSURE MALTH <= MSALTH  
  CALL GETSIN(MALTH, 0,MSALTH,'BAD NUMBER OF PENS ---')  
  CALL GETSIN(NAMFUN, +1,-1,'EXTRA PLOTTER SPEC ---')  
  IF(MCFIRM.NE.0) WRITE(6,615) MALTH  
615 FORMAT(' PLOTTER. PENS. '.11)  
  IF(NAMFUN.NE.'NONE') GO TO 900
```

C

C

C EXIT

C

```
900 KMDP = '  
  RETURN  
  END
```


DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KNDPO1
002

INCLUDE KOMNET.LIST & COMMON CONTROL NETWORK
INCLUDE WINDEF.LIST & DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE NULCST & DEFINE NULL CHARACTER STRING

C
C
C LOCAL DECLARATIONS
C -----
C

REAL RLTEMP(2) & TEMPORARY
INTEGER INTEMP & TEMPORARY
INTEGER KNTMP & TEMPORARY
INTEGER JDSCRIB; & DESCRIPTION (54 CHARACTERS)
INTEGER NQSW & TYPE COORD PRESENT FOR QUERY PT: 'NUL'/'EARTH'/'SCAN'

C
C
C PROCEDURE
C -----
C

CALL TRACE

C
C
C CHECK IF EXPLICIT 'POINT' COMMAND (KOMD IS P(0))
C

IF((KOMD.EQ.'P').OR.
& (KOMD.EQ.'PO').OR.
& (KOMD.EQ.'PO1')) GO TO 110

C
C
C CHECK IF IMPLICIT 'POINT' COMMAND (KOMD IS POINT NUMBER)
C

CALL UNGETS
INTEMP=0 & NOT VALID POINT NUMBER
CALL GETSIN(INTEMP, -99999,99999,NULCST)
IF(INTEMP.EQ.0) GO TO 999
CALL UNGETS

C
C
C INITIALIZE
C

110 IF(NETH1.LT.350) GO TO 120
 CALL MDEFATL('MORE THAN 350 POINTS IN NETWORK')
 GO TO 900
120 NETH:=NETH1+1 & SET NODE POINTER TO NEXT AVAILABLE NODE
 NQSW='NUL' & NOT A QUERY POINT
 NOSAVE=NDTOTL

C
C
C POINT NUMBER
C

NETPT(NETH1):=999999
CALL GETSIN(NETPT(NETH1), -9999,999, '*BAD POINT NUMBER --')
IF(NETPT(NETH1).EQ.0) CALL WARN('BAD POINT NUMBER --')
IF(NETPT(NETH1).GE.-999) GO TO 300
IF(INWINDOM.OF.0) GO TO 250
CALL MOWARN('NETWORK NOT YET ADJUSTED')
GO TO 300

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KRDP01
003

```

250 IF(INCTLPT.LT.8) CALL MDWARN( 'FEWER THAN 8 CONTROL POINTS')
    IF(PCTCTL.LT.40.) CALL MDWARN( 'LESS THAN 40% COVERAGE')
    IF(RMSNET.GT.150.) CALL MDWARN( 'RMS ERROR > 150 METERS')
C
C
C SCANNER (OSFC-ADJUSTED) COORDINATES
C
300 KHTEMP='ABSENT'
    CALL GETSKH(KHTEMP,(1), NULCST)
    IF((KHTEMP.EQ.'S').OR.(KHTEMP.EQ.' ')) GO TO 300      & '(SCAN)'
    CALL UNGETS      & SO WE CAN GET FIELD AGAIN WITH A DIFFERENT FORMAT
    IF(NETPT(NETHI).GE.-999) GO TO 320      & NOT A QUERY POINT
    NQSH='EARTH'      & EARTH COORDINATES GIVEN FOR QUERY POINT
    IF(KHTEMP.GT.0) GO TO 340 & ALPHA COORDINATE SYSTEM (LINE & SAMPLE OMITTED)
    NQSH='SCAN'      & SCANNER COORDINATES GIVEN FOR QUERY POINT
320 CALL GETSRL(ADJNET(WLIN,NETHI),
    * 1..-1500..+4500..'BAD LINE --') & REALS OK
    CALL GETSRL(ADJNET(WSAM,NETHI),
    * 1..0..4000..'BAD SAMPLE --') & REALS OK
    IF((NETPT(NETHI).LT.-999).AND.(NOSAVE.NE.NDTOTL)) GO TO 800
C
C
C CHECK EARTH COORDINATE SYSTEM
C
340 KORDSY='ABSENT'
    CALL GETSKH(KORDSY,(3), NULCST)
    IF(KORDSY.EQ.'DEG') GO TO 400
    IF(KORDSY.EQ.'KM ') GO TO 800
    IF(KORDSY.EQ.'MET') GO TO 810
    CALL WARN( 'BAD EARTH COORDINATE SYSTEM --')
    GO TO 800
C
C
C GEOGRAPHIC COORDINATES (DEGREES)
C
400 IF(NQSH.EQ.'SCAN') GO TO 470      & QUERY POINT. SCANNER COORD GIVEN
    CALL GET9SX(GEDNET(WLAT,NETHI), 1..20..80..'BAD LATITUDE --')
    CALL GET5SX(GEDNET(WLON,NETHI), 1..80..180..'BAD LONGITUDE --')
    IF(NOSAVE.NE.NDTOTL) GO TO 800
    IF(NQSH.EQ.'EARTH') GO TO 480      & QUERY POINT. EARTH COORD GIVEN
    IF(MCFIRM.EQ.0) GO TO 900
430 CALL GETDES
    WRITE(6,445)
    & NETPT(NETHI),
    & ADJNET(WLIN,NETHI),ADJNET(WSAM,NETHI),
    & GEDNET(WLAT,NETHI),GEDNET(WLON,NETHI),JDESCR
445 FORMAT:
    & ' POINT..15.
    & ' SCAN..F8.2..'F8.2.
    & ' DEG..F10.5..'F10.5.A3.8A8)
    GO TO 900
470 CALL G4C(GEDNET(WLAT,NETHI),GEDNET(WLON,NETHI),
    * ADJNET(WLIN,NETHI),ADJNET(WSAM,NETHI))
    GO TO 430
480 CALL A4C(ADJNET(WLIN,NETHI),ADJNET(WSAM,NETHI),
    * GEDNET(WLAT,NETHI),GEDNET(WLON,NETHI))

```

00 TO 430

C
 C
 C
 C

UTM COORDINATES

```

800 CF=1E+3      & CONVERT FROM KILOMETRES TO METRES
      00 TO 820
810 CF=1.        & METRES -- NO CONVERSION NEEDED
820 IF(UTMCHD.EQ.0) CALL MDHARN( 'NO UTM ZONE DEFINED')
      CALL GETSRL(RLTEMP(1), CF,0.,1E+6,'BAD EASTING --')
      CALL GETSRL(RLTEMP(2), CF,0.,9E+6,'BAD NORTHING --')
      IF(INDSAVE.NE.NDTOTL) 00 TO 833
      CALL 040(GEDNET(WLAT,NETH1),GEDNET(WLON,NETH1),
      - RLTEMP(1),RLTEMP(2),UTMCHD)
      IF(INDSW.EQ.'EARTH') 00 TO 880 & QUERY POINT, EARTH COORD GIVEN
      IF(MCFIRM.EQ.0) 00 TO 900
830 CALL GETDES
      WRITE(8,845)
      & NETPT(NETH1),
      & ADJNET(WLIN,NETH1),ADJNET(WSAM,NETH1),
      & RLTEMP(1),RLTEMP(2),JDESCR
845 FORMAT(
      & ' POINT, .15,
      & ' SCAN, .F8.2, .F8.2,
      & ' KM, .-3P.F9.4, .F9.4.A3.8A6)
      00 TO 900
870 IF(INDSAVE.NE.NDTOTL) 00 TO 880
      CALL 040(GEDNET(WLAT,NETH1),GEDNET(WLON,NETH1),
      - ADJNET(WLIN,NETH1),ADJNET(WSAM,NETH1))
      CALL 040(RLTEMP(1),RLTEMP(2),
      - GEDNET(WLAT,NETH1),GEDNET(WLON,NETH1),UTMCHD)
      00 TO 830
880 CALL 440(ADJNET(WLIN,NETH1),ADJNET(WSAM,NETH1),
      - GEDNET(WLAT,NETH1),GEDNET(WLON,NETH1))
      00 TO 830
  
```

C
 C
 C
 C

NOT A VALID CONTROL/CHECK POINT -- SET NODE POINTER BACK

```

800 NETH1=NETH1-1
      00 TO 990
  
```

C
 C
 C
 C

KEEP COUNT OF CONTROL POINTS

```

900 IF(NETPT(NETH1).LT.-999) 00 TO 800 & QUERY POINT
      IF(NETPT(NETH1).GT.0) NCTLPT=NCTLPT+1
  
```

C
 C
 C
 C

NORMAL RETURN

```

990 KOND=' '
999 RETURN
  
```

C
 C

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KNDPOI
005

C
C
C

```
SUBROUTINE GETDES  
DO 150 NMD=1,9  
150 JDESCR(NMD)=  
IF(MBATCH.EQ.0) GO TO 900  
CALL GETSKH(JDESCR(2),(48), NULCST)  
IF(JDESCR(2).NE.' ') JDESCR(1)='.'  
CALL GETSIN(TEMP, +1,-1,'EXTRA POINT SPEC --')  
900 RETURN
```

C
C

END

SUBROUTINE KHOPK(8 'POKE' AT (CHANGE) VARIABLE IN LABELLED COMMON
U KOMD) 8 1: FIRST 3 CHARS OF COMMAND 0: SPACES

C
C
C
C HISTORY
C -----

C
C E H SCHLOSSER LEC 10/24/79 REQUIREMENTS
C E H SCHLOSSER LEC 10/25/79 DESIGN/CODE/TEST
C
C

C
C METHOD
C -----

C
C CHECK NAME OF COMMON AND LOCATION FOR VALIDITY.
C IF INVALID, ISSUE DIAGNOSTIC(S). ELSE CHANGE & PRINT CONTENTS
C OF REQUESTED COMMON LOCATION IN ALL OF THE FOLLOWING FORMATS:
C INTEGER, REAL, CHARACTER STRING
C A VARIABLE ENCLOSED IN QUOTES IS ALWAYS INTERPRETED AS
C A CHARACTER STRING.
C
C

C
C MACHINE-DEPENDENT CODE
C -----

C
C DIMENSION & FORMAT SPECIFICATIONS ASSUME 8 CHARACTERS PER INTEGER.
C
C

C
C EXTERNAL REFERENCES
C -----

C
C MDCLRW 8 CLEAR 'WARNING' DIAGNOSTICS
C GETSKH 8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
C GETSIN 8 GET INTEGER DATA FIELD FROM UNIT 5
C DCODE 8 DECODE NUMERIC CHARACTER STRING
C HOWARN 8 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
C MOVCS 8 MOVE CHARACTER STRING
C INTEGER LCHREQ 8 LOCATE CHARACTER EQUAL SEARCH CHARACTER
C DOUBLE PRECISION CBS4CS 8 VARIABLE-LENGTH CST FOR FIXED-LENGTH CST
C
C

C
C EXCEPTIONS
C -----

- C
C 1. ALL SPECIFICATIONS ARE CHECKED FOR VALIDITY AND RESULT
C IN WARNING DIAGNOSTICS IF BAD.
C
C 2. KOMD IS NOT CHECKED FOR VALIDITY.
C
C

C
C GLOBAL DECLARATIONS
C -----

C
C COMMON/KOMALT/KOMALT(1) 8 COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS
C COMMON/KOMDET/KOMDET(1) 8 COMMON DETECTION FILE WINDOW PKTS & DATES
C

COMMON/KONFIT/KONFIT(1) : COMMON ADJUSTMENT/REGISTRATION PARAMETERS
 COMMON/KONIRT/KONIRT(1) : COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
 COMMON/KONINH/KONINH(1) : COMMON INPUT WINDOW PACKETS
 COMMON/KONKLS/KONKLS(1) : COMMON CLASSIFICATION INFO
 INCLUDE KONLOG.LIST : COMMON LOG FILE BUFFER, I/O PKT. POINTERS
 COMMON/KONLU2/KONLU2(1) : COMMON POINTERS/FLAGS FOR UNIT 2
 COMMON/KONLU3/KONLU3(1) : COMMON POINTERS/FLAGS FOR UNIT 3
 COMMON/KONLU5/KONLU5(1) : COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
 COMMON/KONL2N/KONL2N(1) : COMMON I/O PACKETS FOR DETECTION FILES (21-24)
 COMMON/KONNER/KONNER(1) : COMMON ERTS SCENE PARAMETERS
 COMMON/KONNET/KONNET(1) : COMMON CONTROL NETWORK COORDINATES
 COMMON/KONOWH/KONOWH(1) : COMMON OUTPUT WINDOW PACKETS
 COMMON/KONSLM/KONSLM(1) : COMMON SPECTRAL LIMITS
 COMMON/KONSYM/KONSYM(1) : COMMON SYMBOL TABLE
 COMMON/KONTBL/KONTBL(1) : COMMON MULTI-PURPOSE TABLE
 INCLUDE KONXQT.LIST : COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
 INCLUDE NULCST.LIST : DEFINE NULL CHARACTER STRING

C
 C
 C LOCAL DECLARATIONS
 C -----
 C

INTEGER NAMCOM : NAME OF COMMON TO 'POKE' AT
 INTEGER LOCPOK : LOCATION OF VARIABLE IN COMMON TO 'POKE' AT
 INTEGER KHPOKE(2) : CHARACTER STRING VALUE TO INSERT IN COMMON
 INTEGER KODTYP : CODE TYPE WHEN DECODING KHPOKE
 INTEGER INPOKE : INTEGER VALUE TO INSERT IN COMMON
 REAL RLPOKE : REAL VALUE TO INSERT IN COMMON
 REAL ZERO/.0/ : ZERO!
 LOGICAL POKED : TRUE IF POKE CALLED, FALSE IF NOT

C
 C
 C PROCEDURE
 C -----
 C

CALL TRACE

C
 C
 C GET NAME OF COMMON
 C

CALL MDCLRW(NULCST)
 NAMCOM=' ??? '
 CALL GETSKH(NAMCOM.(6), NULCST)

C
 C
 C GET/CHECK LOCATION OF VARIABLE TO 'POKE' AT
 C

CALL GETSIN(LOCPOK, 1.2500, 'BAD POKE LOCATION --')

C
 C
 C GET/DECODE VALUE TO INSERT
 C

KHPOKE(1)=' ??? '
 CALL GETSKH(KHPOKE.(12), NULCST)
 CALL DCODE(INPOKE,RLPOKE,KODTYP)

DAN PACKAGE APPENDIX H
 COMMAND ROUTINES

KHPOKE
 004

```

    POKED=.TRUE.
    RLTEMP=COM(LOCPOK)+ZERO      & FORCE NORMALIZATION IF NOT REAL
    NFORMT=1
    IF(ABS(RLTEMP).LT.1.0E+8) NFORMT=2
    IF(ABS(RLTEMP).LT.1.0 ) NFORMT=3
    IF(NFORMT.EQ.1) WRITE(6,
    & 181) NAMCOM,LOCPOK,KOM(LOCPOK),RLTEMP,KOM(LOCPOK)
    181 FORMAT(1X,A6,'('',J4,'') '.110.1X,F14.0,2X,A6)
    IF(NFORMT.EQ.2) WRITE(6,
    & 182) NAMCOM,LOCPOK,KOM(LOCPOK),RLTEMP,KOM(LOCPOK)
    182 FORMAT(1X,A6,'('',J4,'') '.110.1X,F14.6,2X,A6)
    IF(NFORMT.EQ.3) WRITE(6,
    & 183) NAMCOM,LOCPOK,KOM(LOCPOK),RLTEMP,KOM(LOCPOK)
    183 FORMAT(1X,A6,'('',J4,'') '.110.1X,F14.12,2X,A6)
    WRITE(6,185)
    185 FORMAT('      CHANGED TO')
    IF(KODTYP.EQ.'IN') KOM(LOCPOK)=INPOKE
    IF(KODTYP.EQ.'RL' .OR.
    & KODTYP.EQ.'FR' .OR.
    & KODTYP.EQ.'SX') COM(LOCPOK)=RLPOKE
    IF(KODTYP.NE.'ERR') GO TO 200
    LOCH1 = LCHREQ(KHPOKE(1),1.6,'...')
    IF(LOCH1.EQ.0) GO TO 190
    LOCH2 = LCHREQ(KHPOKE(1),LOCH1+1.6-LOCH1,'...')
    IF(LOCH2.EQ.0) LOCH2 = 7
    CALL MOVCS(T(KHPOKE(1)),1.6,
    & KHPOKE(1),LOCH1+1,LOCH2-(LOCH1+1),' ')
    190 KOM(LOCPOK) = KHPOKE(1)
    200 RLTEMP=COM(LOCPOK)+ZERO      & FORCE NORMALIZATION IF NOT REAL
    NFORMT=1
    IF(ABS(RLTEMP).LT.1.0E+8) NFORMT=2
    IF(ABS(RLTEMP).LT.1.0 ) NFORMT=3
    IF(NFORMT.EQ.1) WRITE(6,
    & 181) NAMCOM,LOCPOK,KOM(LOCPOK),RLTEMP,KOM(LOCPOK)
    IF(NFORMT.EQ.2) WRITE(6,
    & 182) NAMCOM,LOCPOK,KOM(LOCPOK),RLTEMP,KOM(LOCPOK)
    IF(NFORMT.EQ.3) WRITE(6,
    & 183) NAMCOM,LOCPOK,KOM(LOCPOK),RLTEMP,KOM(LOCPOK)
    RETURN
  END
  
```


DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMPOL
002

C
C LOCAL DECLARATIONS

C -----

C

INTEGER KHTEMP & TEMPORARY
INTEGER NPCHAN & POLAR CHANNEL NUMBER
REAL POAIN(2),PBIAS(2) & TEMP STORAGE FOR GAINS/BIASES AS RL NUMBERS
INTEGER NAME(2) /'RADIUS','ANGLE'/ & INTUITIVE NAMES FOR POLAR.1 & .2

C

C PROCEDURE

C -----

C

CALL TRACE

C

C RETRIEVE CURRENT POLAR GAINS/BIASES

C

POAIN(1)=FLOAT(NRTG12(1))/2.**12
POAIN(2)=FLOAT(NRTG12(2))/2.**12
PBIAS(1)=FLOAT(NRTB24(1))/2.**24
PBIAS(2)=FLOAT(NRTB24(2))/2.**24

C

C GET/CHECK POLAR TRANSFORMED CHANNEL NUMBER

C

NPCHAN=0
CALL GETSIN(NPCHAN, 1.2,'BAD POLAR CHANNEL NUMBER --')
IF(NPCHAN.EQ.0) GO TO 850 & NONE OR INVALID

C

C GET TRANSFORMATION COEFFICIENT TYPE

C

300 KHTEMP=' NUL '
CALL GETSKH(KHTEMP,(3), NULCST)

C

C GET/CHECK/CONFIRM GAIN

C

400 IF(KHTEMP.NE.'GAI') GO TO 500
CALL GETSRL(POAIN(NPCHAN), 1.,-128.,+128.,'BAD GAIN --')
NRTG12(NPCHAN)=POAIN(NPCHAN)*2.**12
IF(NCFIRM.NE.0) WRITE(6,425) NPCHAN,NAME(NPCHAN),POAIN(NPCHAN)
425 FORMAT(' POLAR, '.11,' ('.A6.'). GAIN, '.F9.3)
GO TO 300

C

C GET/CHECK/CONFIRM BIAS

C

500 IF(KHTEMP.NE.'BIA') GO TO 600
CALL GETSRL(PBIAS(NPCHAN), 1.,-100.,+100.,'BAD BIAS --')
NRTB24(NPCHAN)=PBIAS(NPCHAN)*2.**24
IF(NCFIRM.NE.0) WRITE(6,525) NPCHAN,NAME(NPCHAN),PBIAS(NPCHAN)
525 FORMAT(' POLAR, '.11,' ('.A6.'). BIAS, '.F9.3)
GO TO 300

C

DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KNDPOL
003

```
C
C FLAG BAD SPECIFICATION
C
  800 IF(KHTEMP.EQ.' NUL') GO TO 900
      CALL WARN$( 'BAD POLAR SPECIFICATION --')
C
C CONFIRM GAINS AND BIASES
C
  850 IF(MCFIRM.EQ.0) GO TO 900
      DO 860 NPCHAN=1,2
          WRITE(6,425) NPCHAN,NAME(NPCHAN),POAIN(NPCHAN)
          WRITE(6,525) NPCHAN,NAME(NPCHAN),PBIAS(NPCHAN)
  860 CONTINUE
C
C NORMAL RETURN
C
  900 KOND='
      RETURN
      END
```


C GLOBAL DECLARATIONS

C -----

C

INCLUDE KONXQT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST & DEFINE NULL CHARACTER STRING

C

C

C LOCAL DECLARATIONS

C -----

C

INTEGER NAMFUN & NAME OF PRINTER FUNCTION BEING SPECIFIED

C

C

C PROCEDURE

C -----

C

CALL TRACE

C

C

C CHECK IF COMMAND IS NOW LEGAL

C

IF(NHNDOW.LT.+1) GO TO 100
CALL HDWARN:
= 'PRINTER COMMAND CANNOT BE USED AFTER FIRST WINDOW IS PROCESSED')
GO TO 900

C

C

C GET/CHECK PRINTER FUNCTION TO BE SPECIFIED

C

100 NAMFUN=' NUL'
CALL GETSKH(NAMFUN,(3), NULCST)
IF(NAMFUN.EQ.'INC') GO TO 200
IF(NAMFUN.EQ.'PAO') GO TO 300
IF(NAMFUN.EQ.'DEV') GO TO 400
IF(NAMFUN.EQ.'FIL') GO TO 500
IF(NAMFUN.EQ.'CON') GO TO 600
IF(NAMFUN.EQ.' NUL') GO TO 200

C

C

C FLAG BAD FUNCTION & DRAIN SPECS

C

CALL WARN\$('BAD PRINTER FUNCTION --')
150 NAMFUN=' NUL'
CALL GETSKH(NAMFUN,(3), NULCST)
IF(NAMFUN.NE.' NUL') GO TO 150

C

C

C GET/CHECK/CONFIRM LINES & COLUMNS PER INCH

C

200 CALL GETSIN(LINCH, 2.20,'BAD LINES/INCH --')
CALL GETSIN(KINCH, 2.20,'BAD COLUMNS/INCH --')
CALL GETSIN(NAMFUN, +1,-1,'EXTRA PRINTER SPEC --')
YINCH=LINCH
XINCH=KINCH
IF(INCFIRM.NE.0) WRITE(6,215) LINCH,KINCH
215 FORMAT(' PRINTER. INCH. '.13.' LINES. '.13.' COLUMNS')

DAH PACKAGE APPENDIX H
COMMAND ROUTINES

KNDPRI
... 003

```
      IF(NAMFUN.NE.' NUL') GO TO 900
C
C
C GET/CHECK/CONFIRM LINES & COLUMNS PER PAGE
C
300 CALL GETSIN(LPAGE, 40,99,'BAD LINES/PAGE --')
    CALL GETSIN(KPAGE, 40,1000,'BAD COLUMNS/PAGE --')      @ 133-1000 FOR FILM
    IF(KPAGE.GT.132) CALL WARNS('BAD COLUMNS/PAGE --')
    CALL GETSIN(NAMFUN, +1,-1,'EXTRA PRINTER SPEC --')
    IF(MCFIRM.NE.0) WRITE(6,315) LPAGE,KPAGE
315 FORMAT(' PRINTER. PAGE. '.12.' LINES. '.13.' COLUMNS')
    IF(NAMFUN.NE.' NUL') GO TO 900
C
C
C GET/CHECK/CONFIRM DEVICE TYPE MNEMONIC
C
400 CALL GETSKH(MNEMON,(6), NULCST)
    IF(MNEMON.EQ.' ') CALL WARNS('BAD DEVICE SPECIFICATION --')
    CALL GETSIN(NAMFUN, +1,-1,'EXTRA PRINTER SPEC --')
    IF(MCFIRM.NE.0) WRITE(6,415) MNEMON
415 FORMAT(' PRINTER. DEVICE. '.A6)
    IF(NAMFUN.NE.' NUL') GO TO 900
C
C
C GET/CHECK/CONFIRM NUMBER OF FILE(S)
C
500 MALTH=MIND(MALTH,MSALTH)      @ INSURE MALTH <= MSALTH
    CALL GETSIN(MALTH, 0,MSALTH,'BAD NUMBER OF PRINT FILES --')
    CALL GETSIN(NAMFUN, +1,-1,'EXTRA PRINTER SPEC --')
    IF(MCFIRM.NE.0) WRITE(6,515) MALTH
515 FORMAT(' PRINTER. FILES. '.I1)
    IF(NAMFUN.NE.' NUL') GO TO 900
C
C
C GET/CHECK/CONFIRM PRINTER CONTROL
C
600 KONPR=KONPRT
    KONAME=KONPRT
    CALL GETSKH(KONPR,(3), NULCST)
    IF(KONPR.EQ.'AUT') KONAME='AUTO'
    IF(KONPR.EQ.'MAN') KONAME='MANUAL'
    IF(KONPR.EQ.'NON') KONAME=' NUL'
    IF(KONAME.EQ.KONPRT) CALL WARNS('BAD CONTROL SPEC --')
    IF(KONAME.NE.KONPRT) KONPR=KONPR
    CALL GETSIN(NAMFUN, +1,-1,'EXTRA PRINTER SPEC --')
    IF(MCFIRM.NE.0) WRITE(6,615) KONAME
615 FORMAT(' PRINTER. CONTROL. '.A6)
C
C
900 KOND='
    RETURN
    END
```


DAN PACKAGE APPENDIX M
 COMMAND ROUTINES

KNDRAD
 002

```

C -----
C
C     INTEGER KHTEMP      & TEMPORARY
C     INTEGER JPRFMT(8)  & DYNAMIC PRINT FORMAT SPECIFICATION
C     INTEGER MASKOL     & NUMBER OF COLUMN TO START MASKING OUT WITH 'T' FORMAT
C
C PROCEDURE
C -----
C
C     CALL TRACE
C
C ANY SPECIFICATIONS FROM USER?
C
C     KHTEMP=' NUL '
C     CALL GETSKN(KHTEMP,1), NULCST)
C     IF(KHTEMP.EQ.' NUL ') GO TO 800      & NO SPECS. SO CONFIRM CURRENT ONES
C     CALL UNCLTS
C
C GET/CHECK RADIANCE SPECIFICATIONS FOR LIMIT CHANNEL(S)
C
C     IF(NLIMCH.NE.0) GO TO 500
C     CALL NOWARN(
C     - 'NO VALID CHANNEL COMMAND BEFORE RADIANCE COMMAND')
C     GO TO 800
C 500 DO 550 N=1,NLIMCH
C     CALL GETSIN(LCVLO(N), 0.127,'*BAD MIN RADIANCE --')
C     CALL GETSIN(LCVHI(N), 127,'*BAD MAX RADIANCE --')
C 550 CONTINUE
C 500 CALL GETSIN(I'AMP, -1,-1,'EXTRA RADIANCE SPECIFICATION --')
C
C CONFIRM RADIANCE LIMITS
C
C 600 IF(MCFIRM.EQ.0) GO TO 800
C     MASKOL=10+9*MAX(NLIMCH,1)
C     ENCODE(48,825,JPRFMT) MASKOL
C 625 FORMAT('11M RADIANCE. 5(13.1M,13.2M,1.'). & WRITE ALL RADIANCE LIMITS
C     & 'T'.J2,'.45X') & THEN BLANK OUT UNUSED ONES
C     WRITE(6,JPRFMT)
C     & LCVLO(1),LCVHI(1),LCVLO(2),LCVHI(2),LCVLO(3),LCVHI(3),
C     & LCVLO(4),LCVHI(4),LCVLO(5),LCVHI(5)
C
C NORMAL RETURN
C
C 800 KOND=' '
C     RETURN
C     END
  
```


DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KHDRAN
001

(NOT IMPLEMENTED)

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KNDREN
001

SUBROUTINE KNDREN(@ GET/CHECK NEW WINDOW SEQUENCE NUMBER
U KOKD) @ I: FIRST 3 CHARS OF COMMAND @: SPACES

```
C
C
C
C HISTORY
C -----
C
C     E M SCHLOSSER      LEC      10/20/75      ORIGINAL CODE
C     E M SCHLOSSER      LEC      07/17/78      DELETE RET K
C     M A TOMPKINS       LEMSCO   09/27/80      UPGRADE DOCUMENTATION
C
C METHOD
C -----
C
C     IF NWNDOW (<) INITIAL STATE THEN GET NEW WINDOW NUMBER FROM UNIT 5.
C     IF SPECIFIED, AND CONFIRM. INITIALIZE NWNDOW TO NEW VALUE
C     MAINTAINING PRESENT STATE.
C
C MACHINE-DEPENDENT CODE
C -----
C
C     NONE.
C
C EXTERNAL REFERENCES
C -----
C
C     GETSIN      @ GET INTEGER DATA FIELD FROM UNIT 5
C     WARNS      @ PROCESS WARNING DIAGNOSTIC FOR UNIT 5
C
C EXCEPTIONS
C -----
C
C     1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
C        'WARNING' DIAGNOSTICS:
C         1 <= WINDOW SEQUENCE NUMBER <= 999
C
C     2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC MESSAGE.
C
C GLOBAL DECLARATIONS
C -----
C
C     INCLUDE KOMXQT.LIST      @ COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C
C LOCAL DECLARATIONS
C -----
C
C     INTEGER ITEMP      @ ABSOLUTE OF CURRENT WINDOW NUMBER
C
C
```

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDREN
002

C PROCEDURE

C -----

C

CALL TRACE

C

IF(NWINDOW.EQ.0) GO TO 800
ITEMP=ABS(NWINDOW)
CALL GETSIN(ITEMP, 1.999, '*BAD WINDOW NUMBER --')
IF(INCFIRM.NE.0) WRITE(6,125) ITEMP
125 FORMAT(' RENUMBER. '.13)

C

C

ALTER WINDOW TO ITEMP KEEPING ORIGINAL SIGN (STATE).

C

NWINDOW=ISIGN(ITEMP.NWINDOW)
GO TO 900
800 CALL WARNS('INVALID DEFAULT COMMAND --')
900 KOND=''
RETURN
END

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KHORE'S
001

(NOT IMPLEMENTED)

DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KHOSCA
002

```
C
C
C PROCEDURE
C -----
C
C      CALL TRACE
C
      IRFABS=IABS(IRFD)
      FRTEMP=1./((FLOAT(IRFABS)+.00001)
      CALL GET5FR(FRTEMP, 1.,1./1000000.,1./20000.,'BAD SCALE ---')
      IRFABS=(1./FRTEMP)+.5
100 CALL GETSIN(DDUMMY, +1,-1,'EXTRA SCALE SPECIFICATION ---')
      IF(MCFIRM.EQ.0) GO TO 900
      IF(IRFABS.GT.99999) GO TO 140
      WRITE(6,125) IRFABS
125 FORMAT(' SCALE, 1/''.15)
      GO TO 900
140 WRITE(6,145) IRFABS
145 FORMAT(' SCALE, 1/''.16)
C
C
C IF SCALE WAS CHANGED. FLAG AS UNCALIBRATED (NEGATIVE)
C
900 IF(IABS(IRFD).NE.IRFABS) IRFD=-IRFABS
      KOND='
      RETURN
      END
```


DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDSCE
002

```

      INTEGER INTMP      8 TEMP
C
C
C PROCEDURE
C -----
C
C      CALL TRACE
C
C
C GET SCENE NUMBER
C
      JSCENE(1)=' NUL'
      CALL GETSKH(JSCENE,(1), NULCST)
      IF(JSCENE(1).EQ.' NUL') GO TO 300
C
C
C DECODE PRE-JAN78 SCENE NUMBER (3 DIGITS FOR DAYS SINCE LAUNCH)
C
      DECODE(10,115,JSCENE,ERR=140) NERTS
      115 FORMAT(11,13,16)
      NERTS(3)=-NERTS(3)
      GO TO 200
C
C
C DECODE POST-JAN78 SCENE NUMBER (4 DIGITS FOR DAYS SINCE LAUNCH)
C
      140 DECODE(11,145,JSCENE,ERR=180) NERTS
      145 FORMAT(11,14,16)
      NERTS(3)=-NERTS(3)
      GO TO 200
      180 CALL WARN5( 'BAD SCENE NUMBER --')
C
C
C GET OPTIONAL SAMPLES/SCENE (TEMPORARY FOR COMPATIBILITY WITH PRE- 7912 SYNTAX)
C
      200 INTMP=-9999
      CALL GETSIN(INTMP, 700,7000,'BAD SAMPLES/SCENE --')
      IF(INTMP.NE.-9999) NERSAM=INTMP
      IF(NERSAM.LT.900) NERSAM=NERSAM*4
      IF(INTMP.NE.-9999) CALL MONOTE(
      * CBS4IN(NERSAM,5), 'SAMPLES PER SCENE')
      CALL GETSIN(DUMMY, *1,-1,'EXTRA SCENE SPECIFICATION --')
C
C
C CONFIRM SCENE NUMBER
C
      300 IF(MCFIRM.NE.0) WRITE(6,255) NERTS
      255 FORMAT(' SCENE, ',11,J4,'-',1,J5)
      IF(NERTS(3).LT.0) CALL MDWARN(
      * 'ADJUSTMENT WILL NOT INCLUDE NON-LINEAR CORRECTIONS')
C
C
C NORMAL RETURN
C
      KOND=' '
      RETURN
```


DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KHDSCE
003

END

**DAH PACKAGE APPENDIX M
COMMAND ROUTINES**

**KMDSCR
001**

(NOT IMPLEMENTED)

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KNDSHA
002

C
C LOCAL DECLARATIONS
C -----

C
C INTEGER NSCH 8 NUMBER SHARPENING CHANNEL
C INTEGER NCOE 8 NUMBER OF COEFFICIENTS
C INTEGER NSCHAN 8 CHANNEL TO BE SHARPENED
C REAL SFCOEF(5,2) 8 TEMP STORAGE FOR COEFFICIENTS AS REAL NUMBERS

C
C
C PROCEDURE
C -----

C
C CALL TRACE

C
C
C RETRIEVE CURRENT COEFFICIENTS
C

 DO 220 NSCH=1,NERCHA
 DO 210 NCOE=1,2
210 SFCOEF(NSCH,NCOE)=FLOAT(IRSFI2(NSCH,NCOE))/2.**12
220 CONTINUE

C
C
C GET/CHECK CHANNEL NUMBER TO BE SHARPENED
C

 NSCHAN='NONE'
 CALL GETSIN(NSCHAN, 1,NERCHA,'BAD CHANNEL NUMBER --')
 IF(NSCHAN.EQ.'NONE') GO TO 850 8 NONE OR INVALID

C
C
C GET/CHECK AXIS TO BE SHARPENED
C

 ITEMP='NONE'
 CALL GETSKH(ITEMP,(3), NULCST)
 IF(ITEMP.EQ.'SAM') GO TO 400 8 ONLY SHARPENING OF SAMPLE AXIS SUPPORTED
 CALL WARN5('BAD SHARPENING AXIS --')
 GO TO 850

C
C
C GET/CHECK/CONFIRM COEFFICIENTS
C

400 CALL GETSRL(SFCOEF(NSCHAN,1), 1,.-0.1,-0.1,'BAD COEFFICIENT --')
 IRSFI2(NSCHAN,1)=SFCOEF(NSCHAN,1)*2.**12
 CALL GETSRL(SFCOEF(NSCHAN,2), 1,.-0.8,-0.8,'BAD COEFFICIENT --')
 IRSFI2(NSCHAN,2)=SFCOEF(NSCHAN,2)*2.**12
 IF(INCFIRM.NE.0) WRITE(6,525) NSCHAN,(SFCOEF(NSCHAN,NCOE),NCOE=1,2)
525 FORMAT(' SHARPENING. ',11,'. SAMPLE. ',F7.4,'. ',F7.4)
 GO TO 900

C
C
C CONFIRM ALL SHARPENING FILTER COEFFICIENTS
C

850 IF(INCFIRM.EQ.0) GO TO 900
 DO 880 NSCHAN=1,NERCHA
 WRITE(6,525) NSCHAN,(SFCOEF(NSCHAN,NCOE),NCOE=1,2)

DAN PACKAGE APPENDIX H
CONHAND ROUTINES

KND5HA
003

000 CONTINUE
C
C
C NORMAL RETURN
C
000 KOND-
RETURN
END

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDSIZ
002

```
C
  INTEGER NDSAVE          & SAVE AREA FOR CONTENTS OF NDTOTL ON ENTRY
  INTEGER KHTEMP          & TEMPORARY
C
C
C PROCEDURE
C -----
C
C
C   CALL TRACE
C
C
C GET/CHECK SCAN (GSFC-ADJUSTED) COORDINATES
C
  NDSAVE=NDTOTL
  KHTEMP=' NUL '
  CALL GETSKH(KHTEMP,(3), NULCST)
  IF(KHTEMP.EQ.' NUL ') GO TO 400      & NO SPECS -- CONFIRM
  IF(KHTEMP.NE.'SCA') CALL WARN5(' COORDINATE SYSTEM NOT SCAN --')
  IF(KHTEMP.NE.'SCA') GO TO 400
  CALL GETSIN(NERLIN, 2000.4000,'BAD LINES --')
  CALL GETSIN(NERSAM, 2000.4000,'BAD SAMPLES --')
  CALL GETSIN(KHTEMP, +1.-1,'EXTRA SIZE SPECIFICATION --')
C
C
C CONFIRM SIZE COORDINATES
C
  400 IF(MCFIRM.NE.0) WRITE(6,445) NERLIN,NERSAM
  445 FORMAT(' SIZE. SCAN. '.15.''.15)
C
C
C NORMAL RETURN
C
  900 KOMD=' '
  RETURN
  END
```


DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMOSPA
002

```
C
  REAL SPL100      @ LINE SPACING * 100
  REAL SPS100      @ SAMPLE SPACING * 100
  INTEGER ILINSP   @ LINE SPACING
  INTEGER ISAMSP   @ SAMPLE SPACING

C
C PROCEDURE
C -----
C
  CALL TRACE

C
C GET LINE AND SAMPLE SPACING SPECS
C
  SPL100=MSAONH(WLIN,WSP100)
  SPS100=MSAONH(WSAM,WSP100)
  CALL GETSFR(SPL100,
    * 100.,19.99,2000.01,'BAD LINE SPACING --')
  CALL GETSFR(SPS100,
    * 100.,19.99,2000.01,'BAD SAMPLE SPACING --')
  MSAONH(WLIN,WSP100)=SPL100*0.5
  MSAONH(WSAM,WSP100)=SPS100*0.5
  KSYONH(WSP100)='SCA'

C
C FLUSH REMAINING SPECS
C
  CALL GETSIN(IDUMMY,
    * 1.-1,'EXTRA SPACING SPECIFICATION --')

C
C CONFIRM SPACING FOR LINE AND SAMPLE
C
  IF(MCFIRM.EQ.0) GO TO 800
  IF((SPL100/100..EQ.AINT(SPL100/100.)) .AND.
    & (SPS100/100..EQ.AINT(SPS100/100.))) GO TO 500

C
C CONFIRM REAL(S) SPACING
C
  WRITE(6,400) SPL100,SPS100
400 FORMAT(' SPACING. ',2P,F5.2,' LINES. ',F5.2,' SAMPLES')
  GO TO 800

C
C CONFIRM INTEGERS SPACING
C
  500 ILINSP=SPL100/100.
  ISAMSP=SPS100/100.
  WRITE(6,600) ILINSP,ISAMSP
  600 FORMAT(' SPACING. ',12,' LINES. ',12,' SAMPLES')

C
C CLEAR THE COMMAND
C
```

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KNOSPA
003

800 KOND-
C
C
C RETURN TO CALLING ROUTINE
C
900 RETURN
END

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KHOSPH
001

(NOT IMPLEMENTED)


```
INCLUDE KMSYH.LIST      & COMMON SYMBOL TABLE
INCLUDE NULCST.LIST    & DEFINE NULL CHARACTER STRING
INCLUDE MAXINT.LIST    & DEFINE MAXIMUM INTEGER
```

C
C
C
C

LOCAL DECLARATIONS

```
INTEGER NOSAVE          & TEMPORARY SAVE FOR CONTENTS OF NDTOTL ON ENTRY
INTEGER MINSYM,MAXSYM  & MINIMUM, MAXIMUM SYMBOL
INTEGER LENSYM          & LENGTH OF SYMBOL
INTEGER NUM             & NUMBER (RAD/DENSITY/COUNT/CLASS,ETC.) FOR SYMBOL
INTEGER MINNUM,MAXNUM  & MINIMUM, MAXIMUM NUMBER
INTEGER KH3SYM,KH4SYM  & 3RD & 4TH SPECS AS CHARACTER STRINGS
INTEGER IN3NUM,IN4NUM  & 3RD & 4TH SPECS AS NUMBERS
REAL SYNUM             & SYMBOLS PER NUMBER (FOR INTERPOLATION)
```

C
C
C
C

PROCEDURE

CALL TRACE

C
C
C
C

INITIALIZE

NOSAVE=NDTOTL

C
C
C
C

GET OPTIONAL MINIMUM SYMBOL (CHARACTER STRING) FROM 1ST SPEC FIELD

```
MINSYM=' NUL '
CALL GETSKH(MINSYM,(4), NULCST)
IF(MINSYM.EQ.' NUL ') GO TO 850      & NO SPECS
IF((MINSYM.EQ.' : ').OR.           & RESERVED FOR 'NO DATA'
& (MINSYM.EQ.' * ').OR.           & RESERVED FOR PRIMARY TICKS
& (MINSYM.EQ.' + ')).OR.         & RESERVED FOR SECONDARY TICKS
& CALL WARN5( 'BAD FIRST SYMBOL --')
```

C
C
C
C

GET REQUIRED MINIMUM NUMBER (INTEGER) FROM 2ND SPEC FIELD

```
MINNUM=MAX(INT
CALL GETSIN(MINNUM, 0,ISYMH,'BAD FIRST NUMBER --')
```

C
C
C
C

INITIALIZE IMPLICIT MAXIMUM SYMBOL AND NUMBER

```
MAXSYM=MINSYM
MAXNUM=MINNUM
```

C
C
C
C

GET THIRD SPEC FIELD AS BOTH CHARACTER STRING & NUMBER

```
KH3SYM=' NUL '
IN3NUM=-999
```

DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KHOSYN
003

```
      CALL GETSKH(KH3SYH.(4), NULCST)
      CALL UNGETS
      CALL GETSIN(IN3NUM, 0.1SYHMI,NULCST)
C
C
C GET FOURTH SPEC FIELD AS BOTH CHARACTER STRING & NUMBER
C
      KH4SYH=' NUL'
      IN4NUM=-999
      CALL GETSKH(KH4SYH.(4), NULCST)
      CALL UNGETS
      CALL GETSIN(IN4NUM, 0.1SYHMI,NULCST)
C
C
C ASSIGN EXPLICIT MAXIMUM SYMBOL AND NUMBER
C
      IF(KH4SYH.EQ.' NUL') GO TO 120
      MAXSYM=KH3SYH
      MAXNUM=IN4NUM
      GO TO 140
120 IF(KH3SYH.NE.' NUL') MAXNUM=IN3NUM
140 CONTINUE
C
C
C CHECK MAXIMUM SYMBOL AND NUMBER
C
      IF((MAXSYM.EQ.'').OR. 3 RESERVED FOR 'NO DATA'
& (MAXSYM.EQ.'*').OR. 3 RESERVED FOR PRIMARY TICKS
& (MAXSYM.EQ.'*')) 3 RESERVED FOR SECONDARY TICKS
& CALL MDWARN('BAD SECOND SYMBOL')
      IF(MAXNUM.LT.0) CALL MDWARN('BAD SECOND NUMBER')
C
C
C COMPUTE SYMBOLS PER NUMBER
C
      MAXNUM=MIN(MAXNUM,1SYHMI)
      SYINUM=FLOAT((ICE(MAXSYM)-ICE(MINSYM)+1)/
& FLOAT(MAXNUM-MINNUM+1)
& IF(ABS(SYINUM).GT.1.) CALL MDWARN(
& 'MORE THAN 1 SYMBOL PER NUMBER')
C
C
C CHECK FOR EXTRA SPEC FIELDS & FOR DIAGNOSTICS
C
      CALL GETSIN(IN4NUM, +1,-1,'EXTRA SYMBOL SPECIFICATION --')
      IF(INDSAVE.NE.NDTOTL) GO TO 900
C
C
C LOAD SYMBOLS (CHARACTER STRINGS) INTO SYMBOL PART (CHARS 1-4) OF SYMBOL TABLE
C
      ROUND=SIGN(1E-8,SYINUM)
      DO 300 NUM=MINNUM,MAXNUM
          CALL PUTICE(KSYMINUM+1,(1), 3 INTERPOLATE FIRST CHARACTER
& ICE(MINSYM)+FIX(FLOAT(INUM-MINNUM)*SYINUM*ROUND))
          CALL MOVCS(KSYMINUM+1,(2),(3),
& MINSYM,(2),(3), ' ') 3 REPLICATE NEXT 3 CHARS
```


DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KHDSYM
005

C
C
END

C
C LOCAL DECLARATIONS
C -----
C

INTEGER INTENT & TEMPORARY
INTEGER KTBTYP & TABULATION TYPE (3 CHARS)
INTEGER NSYTYP & NUMBER OF SYMBOL TYPE:
 1 = CHARACTER STRING SYMBOL
 2 = INTENSITY SYMBOL
 3 = COLOR SYMBOL
INTEGER NLCST & NUMBER OF LIMIT CHANNEL FOR SYMBOL TYPE
DOUBLE PRECISION MINSYM(3) & CURRENT SYMBOLS (12 CHARS X 3 TYPES)
DOUBLE PRECISION NEXSYM(3) & NEXT SYMBOLS (12 CHARS X 3 TYPES)
INTEGER MINRAD & MINIMUM RADIANCE FOR CURRENT SYMBOL
INTEGER MAXRAD & MAXIMUM RAD/CLA/DEN FOR CURRENT SYMBOL
INTEGER KFRSUB(6),KFRTOT(6) & FREQ SUB-TOTALS, TOTALS

C
C
C PROCEDURE
C -----
C

CALL TRACE

C
C
C CHECK IF KONTBL LOADED WITH FREQUENCY INFO
C

IF((KLSTYP.EQ.0).OR. & NO CLASS TYPE FROM PREVIOUS DISPLAY
& (KTBLTY.NE.'FREQ')) & NO FREQUENCY TABLES LOADED FROM PREVIOUS DISPLAY
& CALL NOWARN('NO PREVIOUS WINDOW TO TABULATE')

C
C
C GET/CHECK TABULATE TYPE
C

KTBTYP='SYM' & DEFAULT IS TABULATION OF CHARACTER STRING SYMBOLS
NSYTYP=MAXINT & NOT YET DEFINED
CALL GETSKM(KTBTYP,(3), NULCST)
IF(KTBTYP.EQ.KLSTYP) NSYTYP=1
IF(KTBTYP.EQ.'SYM') NSYTYP=1
IF((NCOLOR.NE.0).AND.(KTBTYP.EQ.'INT')) NSYTYP=2
IF((NCOLOR.NE.0).AND.(KTBTYP.EQ.'COL')) NSYTYP=3
NLCST=1
IF(NSYTYP.EQ.3) NLCST=2
IF(NSYTYP.EQ.MAXINT) CALL WARN('BAD TABULATE TYPE --')

C
C
C DRAIN SPECS FOR CURRENT CLMHAND
C

CALL GETSINI(TEMP, -1,-1,'EXTRA TABULATE SPECIFICATION --')

C
C
C CHECK FOR DIAGNOSTICS
C

IF(MDATA.NE.0) GO TO 900 & DATA/CHECKOUT MODE
IF(NDTOTL.NE.0) GO TO 800

C
C

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDTAB
003

C USE PREVIOUS WINDOW NUMBER FOR PAGE HEADING

C

 NWTEMP=NNWIND & SAVE CURRENT WINDOW NUMBER
 NNWIND=KTBLNM & WINDOW NUMBER FROM FREQUENCY TABLES

C

C

C PRINT PAGE/WINDOW HEADINGS

C

 WRITE(6,105) NNWIND,MTERAL
105 FORMAT(' WINDOW NUMBER ',J3,6X,'TABULATE',6X,4A6)
 CALL MDUNIT(4,10)
 WRITE(10,105) NNWIND,MTERAL

C

C

C RESTORE CURRENT WINDOW NUMBER

C

 NNWIND=NWTEMP

C

C

C PRINT TABLE HEADINGS

C

 CALL IDERT(6)
 CALL IDERT(10)
 IF(NSATCH.EQ.0)
 & WRITE(6,125) LIMCH(NLCSY), (LIMCH(N),N=1,NLIMCH)
 WRITE(10,125) LIMCH(NLCSY), (LIMCH(N),N=1,NLIMCH)
125 FORMAT:
 & 'SYMBOL RADIANCE FREQUENCY'
 & ' CH'.12.' VALUE '.5(' CH'.12)'
 WRITE(6,135)
 WRITE(10,135)
135 FORMAT(1X)

C

C

C INITIALIZE SUB-TOTALS, TOTALS & LIMITS

C

 DO 340 N=1,5
 KFRSUB(N)=0
 KFRTOT(N)=0

340 CONTINUE

 MINCHV=-MAXINT

 MAXCHV=-MAXINT

 DO 360 N=1,NLIMCH

 MINCHV=MIN0(MINCHV,LCVLO(N))

 MAXCHV=MAX0(MAXCHV,LCVHI(N))

360 CONTINUE

 MINRAD=MINCHV

 CALL MOVCSY(MINSYH(1), (1), (3), ' ', (1), (1), ' ')

 CALL MOVCSY(MINSYH(1), (4), (6), KSYH(MINRAD+1), (1), (1), ' ')

 CALL MOVCSY(MINSYH(1), (10), (3), KSYH(MINRAD+1), (2), (3), ' ')

 CALL GETICE(INTENP, KSYH(MINRAD+1), (5))

 CALL CSTN(MINSYH(2), (1), (12), (INTENP-1)*10.4)

 CALL PUTCHR(MINSYH(2), (5), 'S')

 CALL GETICE(INTENP, KSYH(MINRAD+1), (6))

 CALL KOLR4(MINSYH(3), INTENP)

C

```

C
C PRINT TABULATIONS
C
      DO 800 MAXRAD=MINCHV,MAXCHV
      DO 410 N=1,NLIMCH
          KFRSUB(N)=KFRSUB(N)+KFRSQ(MAXRAD+1,N)
410 CONTINUE
      CALL MOVCST(NEXSYH(1),(1),(3), ' ',(1),(1),' ')
      CALL MOVCST(NEXSYH(1),(4),(6), KSYH(MAXRAD+2),(1),(1),' ')
      CALL MOVCST(NEXSYH(1),(10),(3), KSYH(MAXRAD+2),(2),(3),' ')
      CALL GETICE(INTENP, KSYH(MAXRAD+2),(9))
      CALL CST4IN(NEXSYH(2),(1),(12), (INTENP+1)*10,4)
      CALL PUTCHR(NEXSYH(2),(5), '8')
      CALL GETICE(INTENP, KSYH(MAXRAD+2),(6))
      CALL KOLR4(NEXSYH(3), INTENP)
      IF((KTBTYP.NE.KLSTYP).AND. & AGGREGATE TABULATIONS BY SYMBOL
          & (MAXRAD.LT.LCVH1).AND. & MORE TO GO
          & (NEXSYH(NSYTP).EQ.MINSYH(NSYTP))) & NEXT SYMBOL SAME AS THIS
          & GO TO 800 & DON'T PRINT ENTRY FOR THIS SYMBOL YET
          IF((MAXRAD.LT.LCVLO1).OR.(MAXRAD.GT.LCVH1))
              CALL MOVCST(MINSYH(1),(36), ' ',(1),(1),' ')
          IF(MINRAD.EQ.MAXRAD) GO TO 440
          IF(MBATCH.EQ.0)
              WRITE(6,425) MINSYH(NSYTP),MINRAD,MAXRAD,
              & (KFRSUB(N),N=1,NLIMCH)
              WRITE(10,425) MINSYH(NSYTP),MINRAD,MAXRAD,
              & (KFRSUB(N),N=1,NLIMCH)
425 FORMAT(1X,A9,1X,J3,' - ',J3,516)
              GO TO 480
440 IF(MBATCH.EQ.0)
              WRITE(6,445) MINSYH(NSYTP),MINRAD,(KFRSUB(N),N=1,NLIMCH)
              WRITE(10,445) MINSYH(NSYTP),MINRAD,(KFRSUB(N),N=1,NLIMCH)
445 FORMAT(1X,A9,4X,J3,3X,516)
480 IF(NCISYM.LT.2) GO TO 550
              DO 500 NCHAR=2,NCISYM & OVERPRINT SYMBOLS
                  CALL GETCHR(KHTEMP, MINSYH,(NCHAR-0))
                  IF(KHTEMP.NE.' ') WRITE(10,475) KHTEMP
475 FORMAT(' ',3X,A1)
500 CONTINUE
550 DO 570 N=1,5
              KFRTOT(N)=KFRTOT(N)+KFRSUB(N)
              KFRSUB(N)=0
570 CONTINUE
              MINRAD=MAXRAD+1
              CALL MOVCST(MINSYH(1),(36), NEXSYH(1),(36),' ')
C
C 800 CONTINUE
C
C PRINT TOTALS
C
      IF(MBATCH.EQ.0)
          & WRITE(6,815) (KFRTOT(N),N=1,NLIMCH)
          & WRITE(10,815) (KFRTOT(N),N=1,NLIMCH)
815 FORMAT(' TOTAL ',12X,17,416)
      GO TO 900
  
```

1962-1963

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDTAB
005

```
C
C
C CHECK DIAGNOSTIC COUNTERS
C
  800 IF(INDWARN.EQ.0) GO TO 820
      CALL MDNOTE('PREVIOUS WARNINGS -- NO TABULATION GENERATED')
      IF(MBATCH.EQ.0) WRITE(6,815)
  815 FORMAT('...TRY AGAIN')
      GO TO 890
  820 IF(INDFATL.EQ.0) GO TO 850
      CALL MDNOTE('PREVIOUS FATAL ERRORS -- NO TABULATION GENERATED')
      GO TO 890
  850 IF(INCHECK.EQ.0) GO TO 890
      CALL MDNOTE('CHECKOUT MODE -- NO TABULATION GENERATED')

C
C CLEAR WARNINGS
C
  890 CALL MDCLRW( NULCST)

C
C RETURN
C
  900 KMD=
      RETURN
      END
```


C 3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

C
 C
 C
 C
 C

GLOBAL DECLARATIONS

```

INCLUDE KONXQT.LIST      % COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOHIHW.LIST     % COMMON INPUT WINDOW PACKETS
INCLUDE KONOWH.LIST     % COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDEF.LIST     % DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE NULCST.LIST     % DEFINE NULL CHARACTER STRING
  
```

C
 C
 C
 C
 C

LOCAL DECLARATIONS

```

REAL RLTEMP(2)          % TEMPORARY
INTEGER JTYPE(2)        % TYPE OF TICKS: 'PRIMARY' OR 'SECONDARY'
  
```

C
 C
 C
 C
 C

PROCEDURE

CALL TRACE

C
 C
 C
 C
 C

INITIALIZE

```

CALL SPAN5( 2)          % ALLOW INPUT FOR THIS COMMAND TO SPAN 2 CARDS
KMD='TICK.'
JTYPE(1)='PRIMA'
JTYPE(2)='RY.'
NOD=WTIC                % START WITH PRIMARY TICKS
  
```

C
 C
 C
 C
 C

GET COORDINATE SYSTEM

```

200 KORDSY=KSYOWH(NOD)  % USE OLD COORDINATE SYTEM IF NONE SPECIFIED
CALL GET5KH(KORDSY,(3), NULCST)
IF(KORDSY.GT.0) GO TO 220 % POSITIVE WORD STARTS WITH ALPHA CHARACTER
  
```

C
 C
 C
 C

IF SECONDARY TICKS HAVE NO COORD SYSTEM USE SYSTEM FROM PRIMARY TICKS

```

IF(NOD.EQ.WTIC) GO TO 280 % PRIMARY TICKS
KORDSY=KSYOWH(WTIC)
CALL UNGETS
  
```

C
 C
 C
 C

CHECK COORDINATE SYSTEM

```

220 IF(KORDSY.EQ.'SCA') GO TO 300
IF(KORDSY.EQ.'DEG') GO TO 400
IF(KORDSY.EQ.'MIN') GO TO 410
IF(KORDSY.EQ.'KM ') GO TO 600
IF(KORDSY.EQ.'MET') GO TO 610
  
```

DAH PACKAGE APPENDIX H
COMMAND ROUTINES

KMDTIC
003

```
280 CALL WARN5( 'BAD COORDINATE SYSTEM --')
    GO TO 900
C
C
C SCANNER (LINE-LENGTH ADJUSTED) COORDINATES
C
300 CALL GET5IN(MSAOWH(WLIN.NOD), 2.3000, 'BAD LINES --')
    CALL GET5IN(MSAOWH(WSAM.NOD), 2.4000, '*BAD SAMPLES --')
340 IF(MCFIRM.NE.0) WRITE(6,345)
    & KMD,MSAOWH(WLIN.NOD),MSAOWH(WSAM.NOD),JTYPE
345 FORMAT(1X,A6,' SCAN. ',14,' LINES. ',14,' SAMPLES',2A6)
    GO TO 800
C
C
C GEOGRAPHIC COORDINATES
C
400 CF=1.      & DEGREES -- NO CONVERSION NEEDED
    GO TO 420
410 CF=1./60.   & CONVERT FROM MINUTES TO DEGREES
420 CALL GET5SX(GEDOWH(WLAT.NOD), CF,.001,3,'BAD LATITUDE --')
    CALL GET5SX(GEDOWH(WLON.NOD), CF,.001,3,'BAD LONGITUDE --')
    IF(KORDSY.EQ.'MIN') GO TO 450
    IF(MCFIRM.EQ.0) GO TO 800
    WRITE(6,445) KMD,GEDOWH(WLAT.NOD),GEDOWH(WLON.NOD),JTYPE
445 FORMAT(1X,A6,' DEGREES. ',F7.5,' LAT. ',F7.5,' LON',2A6)
    GO TO 800
450 RLTEMP(1)=GEDOWH(WLAT.NOD)*60.
    RLTEMP(2)=GEDOWH(WLON.NOD)*60.
    IF(MCFIRM.NE.0) WRITE(6,455) KMD,RLTEMP,JTYPE
455 FORMAT(1X,A6,' MINUTES. ',F7.3,' LAT. ',F7.3,' LON',2A6)
    GO TO 800
C
C
C UTM COORDINATES
C
600 CF=1E+3    & CONVERT FROM KILOMETRES TO METRES
    GO TO 620
610 CF=1.      & METRES -- NO CONVERSION NEEDED
620 CALL GET5RL(UTMOWH(WEA.NOD), CF,1E+2,5E+5,'BAD EASTING --')
    CALL GET5RL(UTMOWH(WNO.NOD), CF,1E+2,5E+5,'BAD NORTHING --')
    IF(MCFIRM.NE.0) WRITE(6,645)
    & KMD,UTMOWH(WEA.NOD),UTMOWH(WNO.NOD),JTYPE
645 FORMAT(1X,A6,' KM. ',3P,F6.3,' EAST. ',F6.3,' NORTH',2A6)
C
C
C STORE COORDINATE SYSTEM
C
800 KSYOWH(NOD)=KORDSY
    IF(NOD.NE.WTIC) GO TO 890
    NOD=WTIC+1      & READY FOR SECONDARY TICKS
    KMD=' '        & DON'T PRINT COMMAND WITH SECONDARY TICKS
    JTYPE(1)=' SECON'
    JTYPE(2)=' DARY '
    GO TO 200
890 CALL GET5IN(ITEMP, +1,-1,'EXTRA TICK SPECIFICATION --')
```


DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KMDTIC
004

C
C NORMAL RETURN
C
900 KOND-
RETURN
END

DAN PACKAGE APPENDIX H
COMMAND ROUTINES

KMDTIM
002

INTEGER ITIME(3) 8 HOURS/MINUTES/SECONDS OF CHARGE TIME
REAL FSECS 8 FRACTIONAL SECONDS OF CHARGE TIME

C
C
C
C
C
C
C

PROCEDURE

CALL TRACE

C

CALL GETSIN(NHRS. *1.-1.'NO TIME SPECIFICATION ALLOWED --')
CALL ERDATE(IMDY, IHMS)
CALL MOVCS1(IM.(1).(2). IHMS.(3).(2).)
CALL MOVCS1(IS.(1).(2). IHMS.(5).(2).)
CALL ERSUPS(MS200)
SECS=FLOAT(MS200)/1800000.
CALL RL2ISX(SECS, ITIME, 3, FSECS)
WRITE(6, I45)
& IHMS, IM, IS,
& ITIME, FSECS
I45 FORMAT(' TIME'
& ' (CLOCK) '.A2.' : '.A2.' : '.A2/'
& ' (CHARGE) '.J2.' : '.J2.' : '.J2.F4.3)

C

KOMD=
RETURN
END

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDTIN
001

(NOT IMPLEMENTED)

DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KMDTOL
002

CALL TRACE

C

```
LCVTOL=MAX(0,LCVTOL)
LCVTOL=MIN(6,LCVTOL)
CALL GETSIN(LCVTOL, 0.8,'BAD TOLERANCE --')
CALL GETSIN(1,DUMMY, +1,-1,'EXTRA TOLERANCE SPECIFICATION --')
IF(INCFIRM.NE.0) WRITE(6,125) LCVTOL
125 FORMAT(' TOLERANCE, '.11)
KOND=' '
RETURN
END
```

DAW PACKAGE APPENDIX H
COMMAND ROUTINES

KNDWIN
001

SUBROUTINE KNDWIN: 8 GET/CHECK WINDOW VERTICES
U KOND) 8 1: FIRST 3 CHARS OF COMMAND 0: SPACES

C
C
C
C HISTORY
C -----

C E H SCHLOSSER LEC 08/09/73 NUMERIC OPTION
C E H SCHLOSSER LEC 12/07/75 ALPHANUMERIC COMMAND
C M A TOMPKINS LEMSCO 09/27/80 UPGRADE DOCUMENTATION
C

C
C METHOD
C -----

C UPDATE WINDOW VERTICES FROM UNIT 5. IF SPECIFIED. AND CONFIRM.
C

C
C MACHINE-DEPENDENT CODE
C -----

C USES UNIVAC FORTRAN V RETURN K.
C

C
C EXTERNAL REFERENCES
C -----

C HDWARN 8 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
C UNGETS 8 BACK UP 1 FIELD ON UNIT 5
C GETSKM 8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
C GETSIN 8 GET INTEGER DATA FIELD FROM UNIT 5
C GETSRL 8 GET REAL DATA FIELD FROM UNIT 5
C GETSSX 8 GET SEXAGENARY DATA FIELD FROM UNIT 5
C WARNS 8 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
C GETSIN 8 GET INTEGER DATA FIELD FROM UNIT 5
C SPANS 8 ENABLE SPECIFICATIONS TO SPAN CARDS
C HRVERT 8 WRITE VERTEX COORDINATES FOR WINDOW
C

C
C EXCEPTIONS
C -----

C 1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
C 'WARNING' DIAGNOSTICS:

C -2500 <= SCANNER LINE <= 2500
C -3500 <= SCANNER SAMPLE <= 3500
C -3. <= LATITUDE DEG <= 90.
C -3. <= LONGITUDE DEG <= 180.
C -1E+4 <= EASTING METRES <= 1E+6
C -1E+4 <= NORTHING METRES <= 9E+6
C -800 <= PRINT LINES <= 800
C -800 <= PRINT COLUMNS <= 800
C

C 2. TOO MANY VERTICES GENERATE A 'WARNING' DIAGNOSTIC.
C

DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KNDMIN
002

C
C GLOBAL DECLARATIONS
C -----
C

INCLUDE KOMXQT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMINH.LIST & COMMON INPUT WINDOW PACKETS
INCLUDE KOMOHV.LIST & COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDEF.LIST & DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE NULCST.LIST & DEFINE NULL CHARACTER STRING

C
C LOCAL DECLARATIONS
C -----
C

PARAMETER NODMIN = MVER+1 & NODE MINIMUM
INTEGER KORDSY & COORDINATE SYSTEM
INTEGER JCOMMA & COMMA/BLANK DEPENDING ON POSITION IN PRINT LINE
INTEGER NOD & NODE
INTEGER N & NODE COUNTER
INTEGER KONAME & COORDINATE NAME
REAL CFL & CONVERT FROM XXX TO PRINT LINES
REAL CFC & CONVERT FROM XXX TO PRINT COLUMNS
REAL CF & CONVERSION FACTOR
INTEGER MP & MAXIMUM # OF PLACES

C
C PROCEDURE
C -----
C

CALL TRACE

C
C INITIALIZE
C

CALL SPANS(10) & ALLOW INPUT FOR THIS COMMAND TO SPAN UP TO 10 CARDS
NOD=MVER & SET NODE POINTER BEFORE FIRST VERTEX

C
C CHECK COORDINATE SYSTEM
C

KORDSY=KSYOH(MVER) & USE OLD COORDINATE SYSTEM IF NONE SPECIFIED
CALL GETSK(KORDSY,(3), NULCST)
IF(KORDSY.EQ.'SCA') 00 TO 300 & SCANNER COORDINATES
IF(KORDSY.EQ.'DEG') 00 TO 400 & GEOGRAPHIC COORDINATES IN DEGREES
IF(KORDSY.EQ.'MIN') 00 TO 410 & GEOGRAPHIC COORDINATES IN MINUTES
IF(KORDSY.EQ.'KM') 00 TO 600 & UTM COORDINATES IN KILOMETRES
IF(KORDSY.EQ.'MET') 00 TO 610 & UTM COORDINATES IN METRES
IF(KORDSY.EQ.'PRI') 00 TO 700 & MAP COORDINATES IN PRINT LINES & COLUMNS
IF(KORDSY.EQ.'CM') 00 TO 705 & MAP COORDINATES IN CENTIMETRES
IF(KORDSY.EQ.'INC') 00 TO 710 & MAP COORDINATES IN INCHES
CALL WARN('BAD COORDINATE SYSTEM --')
00 TO 900

C
C SCANNER (LINE-LENGTH ADJUSTED) COORDINATES

DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KNDWIN
003

C

```
300 CALL CKCLOS(MSAOMM,8340,8900)
    CALL GETSIN(MSAOMM(MLIN,NOD), -2500,2500, '*BAD LINE --')
    CALL GETSIN(MSAOMM(MSAN,NOD), -3500,3500, '*BAD SAMPLE --')
    GO TO 300
340 WRITE(8,345)
345 FORMAT(' WINDOW. SCAN. ')
    JCOMMA=', '
    DC 360 N=NODMIN,NOD
    IF(IN.EG,NOD) JCOMMA=', '
    WRITE(8,355) (MSAOMM(I,N),I=1,2),JCOMMA
355 FORMAT(' (3X,15.' LINE. '15.' SAMPLE',A1))
360 CONTINUE
    GO TO 900
```

C

C

C GEOGRAPHIC COORDINATES

C

```
400 KONAME='DEGREE'
    CF=1.      8 DEGREES -- NO CONVERSION NEEDED
    MP=3      8 MAX NO OF SEX'Y PLACES
    GO TO 420
410 KONAME='MINUTE'
    CF=1./60.  8 CONVERT FROM MINUTES TO DEGREES
    MP=2      8 MAX NO OF SEX'Y PLACES
420 CALL CKCLOS(DEDOWH,8440,8900)
    CALL GETSSX(DEDOWH(MLAT,NOD), CF,-3.,80., '*BAD LATITUDE --')
    CALL GETSSX(DEDOWH(MLON,NOD), CF,-3.,180., '*BAD LONGITUDE --')
    GO TO 420
440 WRITE(8,445) KONAME
445 FORMAT(' WINDOW. '.A6,'S. ')
    CALL WRVERT(DEDOWH,CF,CF,'(3X,F9.4,6H LAT. ,F9.4,4H LON,A1)',NOD)
    GO TO 900
```

C

C

C UTM COORDINATES

C

```
600 CF=1E+3    8 CONVERT FROM KILOMETRES TO METRES
    GO TO 620
610 CF=1.      8 METRES -- NO CONVERSION NEEDED
620 CALL CKCLOS(UTMOWH,8640,8970)
    CALL GETSRL(UTMOWH(MEA,NOD), CF,-1E+4,1E+6, '*BAD EASTING --')
    CALL GETSRL(UTMOWH(MNO,NOD), CF,-1E+4,9E+6, '*BAD NORTHING --')
    GO TO 620
640 WRITE(8,645) KORDSY
645 FORMAT(' WINDOW. '.A3,'. ')
    CALL WRVERT(UTMOWH,CF,CF,
    6 '(3X,F9.3,7H EAST. ,F9.3,6H NORTH,A1)',NOD)
    GO TO 900
```

C

C

C PRINT/PLOT COORDINATES

C

```
700 KONAME='PRINT'
    CFL=1.     8 PRINT LINES -- NO CONVERSION NEEDED
    CFC=1.     8 PRINT COLUMNS -- NO CONVERSION NEEDED
```

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KRDWIN
004

```

GO TO 720
705 KONAME=KORDSY
   CFL=0.3937*LINCH      & CONVERT FROM CENTIMETRES TO PRINT LINES
   CFC=0.3937*KINCH     & CONVERT FROM CENTIMETRES TO PRINT COLUMNS
GO TO 720
710 KONAME='INCHES'
   CFL=FLOAT(LINCH)     & CONVERT FROM INCHES TO PRINT LINES
   CFC=FLOAT(KINCH)     & CONVERT FROM INCHES TO PRINT COLUMNS
720 CALL CKCLOS(PPDOWN,9740,9900)
   CALL GETSRL(PPDOWN,HLIN,NOD), CFL,-999..999..'BAD LINE --'
   CALL GETSRL(PPDOWN,ICOL,NOD), CFC,-999..999..'BAD COLUMN --'
GO TO 700
740 WRITE(6,745) KONAME
745 FORMAT(' WINDOW, '.AB,' ')
   IF(KORDSY.NE.'PRI') GO TO 760
   CALL WVERT(PPDOWN,CFL,CFC,
& '13X.F8.0.T8.7H LINE. .F5.0.T18.7H COLUMN.A1').NOD)
GO TO 900
760 CALL WVERT(PPDOWN,CFL,CFC,
& '13X.F8.2.7H LINE. .F8.2.7H COLUMN.A1').NOD)
C
C
C NORMAL RETURN
C
900 KOND='
RETURN
C
C
C
C
C
C
C
SUBROUTINE CKCLOS & CLOSE POLYGON IF NEXT VERTEX IS ABSENT
& INTOWH. & WINDOW PACKET
& S. & CLOSE WITH CONFIRMATION TRANSFER LABEL
& SI & CLOSE WITHOUT CONFIRMATION TRANSFER LABEL
C
DIMENSION INTOWH(2,1)
C
KNTMP=' NUL '
CALL .GETSKH(KNTMP,1), NULCST)
IF(KNTMP.EQ.' NUL ') GO TO 110 & NO MORE SPECS
CALL UNGETS
NOD=NOD+1 & READY TO GET NEXT VERTEX
IF(NOD.LE.MNO) RETURN
CALL MDWARN(' TOO MANY VERTICES -- EXCESS IGNORED ')
NOD=NOD-1
110 IF(NOD.NE.MVER) GO TO 150
NOD=INTOWH(MUSED,MHEAD) & NO VERTICES SPECIFIED -- USE OLD WINDOW
GO TO 900
150 INTOWH(MUSED,MHEAD)=NOD & UPDATE HIGHEST NODE USED
INTOWH(1,MVER)=INTOWH(1,NOD) & CLOSE ORDINATE
INTOWH(2,MVER)=INTOWH(2,NOD) & CLOSE ABSCISSA
900 NXLAST=(NOD-MVER-1)*2 & UPDATE POINTER TO ABSCISSA OF NXT-TO-LST VERTEX

```

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDWIN
005

```
KSYOMH(HVER)=KORDSY  
IF(MCFIRM.EQ.0) RETURN 3  & CLOSED -- DON'T CONFIRM VERTICES  
RETURN 2                  & CLOSED -- CONFIRM VERTICES  
END
```

SUBROUTINE KMDXXX: 8 PROCESS MACRO COMMAND
U KOMD) 8 1: FIRST 3 CHARS OF PROG NAME PLUS '-' 0: SPACES IF VALID

C

C

C

C HISTORY

C -----

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

E M SCHLOSSER LEC 01/17/79 ORIGINAL CODE

C METHOD

C -----

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

GET COMMAND. IF IT IS 8 CHARACTERS OR SHORTER, PREFIX KOMD TO IT.
LOOK FOR A SYMBOLIC ELEMENT WITH THIS NAME FIRST IN THE PRIVATE LIBRARY
(PROGRAM FILE) MACDAM, AND THEN IN THE PUBLIC LIBRARY (PROGRAM FILE) DAM.
IF FOUND, CALL KMXXED AND CHANGE KOMD TO SPACES.

C MACHINE-DEPENDENT CODE

C -----

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

UNIVAC EXEC 8 PROGRAM FILE NAMING CONVENTIONS.

C EXTERNAL REFERENCES

C -----

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

UNGETS 8 BACK UP 1 FIELD ON UNIT 5
PUTCHR 8 PUT CHARACTER INTO CHARACTER STRING
GETSKH 8 GET CHARACTER STRING FIELD FROM UNIT 5
LENCST 8 GET LENGTH OF CHARACTER STRING
MOVCST 8 MOVE CHARACTER STRING
LOCDSF 8 LOCATE DISK SYMBOLIC FILE OR ELEMENT
KMXXED 8 EDIT ACTUAL SPECS INTO MACRO COMMAND DEFINITION

C EXCEPTIONS

C -----

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

1. KOMD IS LEFT UNCHANGED IF ANY OF THE FOLLOWING OCCURS:
LENGTH OF MACRO COMMAND IS < 3 OR > 12 CHARACTERS
NO DISK SYMBOLIC ELEMENT EXISTS FOR MACRO COMMAND
2. SPECIFICATION FIELDS, IF SUPPLIED, ARE CHECKED BY KMXXED.
3. NO OTHER CHECKS ARE MADE.

C GLOBAL DECLARATIONS

C -----

C

C

C

C

C

C

C

C

C

C

C

C

C

C

INCLUDE KOMXQT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST 8 DEFINE NULL CHARACTER STRING

DAM PACKAGE APPENDIX M
 COMMAND ROUTINES

KMDXXX
 002

```

C
C LOCAL DECLARATIONS
C -----
C
      INTEGER MAKOMD(3)      % NAME OF MACRO COMMAND
      INTEGER NAMELT(2)     % NAME OF DISK SYMBOLIC ELEMENT
      INTEGER NAMFIL(2)     % NAME OF DISK SYMBOLIC FILE ('MACDAM' OR 'DAM')
      % '/'                %
      INTEGER LOCFIL        % LOCATION WITHIN DISK SYMBOLIC FILE ( IF > 0 )
      INTEGER LENGTH        % LENGTH OF CHARACTER STRING, EXCL TRAILING BLANKS
C
C
C PROCEDURE
C -----
C
      CALL TRACE
C
C GET 1ST 18 CHARACTERS OF MACRO COMMAND
C
      CALL UNGETS
      CALL PUTCHR(MAKOMD.(18), '?' )      % INITIALIZE MAKOMD TO 18 CHAR LENGTH
      CALL GETSKH(MAKOMD.(18), NULCST)
C
C CHECK LENGTH OF MACRO COMMAND
C
      LENGTH=LENCST(MAKOMD.18)
      IF(LENGTH.LT.3) GO TO 900      % TOO SHORT
      IF(LENGTH.GT.12) GO TO 900    % TOO LONG
C
C CONSTRUCT NAME OF ELEMENT CONTAINING TEXT FOR MACRO COMMAND
C
      NAMELT=KOMD      % 1ST 3 CHARS OF PROGRAM NAME PLUS '-'
      IF(LENGTH.LE.8) CALL MOVCST(NAMELT.(5).(8),
      *                MAKOMD.(1).(8), ' ')      % <KOMD><MAKOMD>
      IF(LENGTH.GT.8) CALL MOVCST(NAMELT.(1).(12),
      *                MAKOMD.(1).(12), ' ')      % <MAKOMD>
C
C CHECK IF ELEMENT WITH TEXT FOR MACRO-COMMAND IS ON DISK
C
      NAMFIL='MACDAM'
      LOCFIL=LOCDSF(NAMFIL.NAMELT,' ')
      IF(LOCFIL.LE.0) NAMFIL='DAM'
      IF(LOCFIL.LE.0) LOCFIL=LOCDSF(NAMFIL.NAMELT,' ')
      IF(LOCFIL.LE.0) GO TO 900      % NO SYMBOLIC ELEMENT FOR MACRO COMMAND
      KOMD='
C
C
C EDIT ACTUAL SPECS INTO MACRO COMMAND DEFINITION
C
      CALL KMXED( MAKOMD.NAMFIL.NAMELT.LOCFIL)
C

```

DAH PACKAGE APPENDIX M
COMMAND ROUTINES

KNOXXX
003

C
C DONE
C
900 RETURN
END

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDZON
002

```
C  
C  
C PROCEDURE  
C -----  
C  
C CALL TRACE  
C  
IF(UTMCHD.EQ.0.) UTMCHD=-35817.      & INVALID (ZONE = 6000)  
OLOCMD=UTMCHD  
INTEMP=(183.-UTMCHD)/6. + .5  
CALL GETSIN(INTEMP, 1.22,'BAD UTM ZONE --')  
UTMCHD=183-6*INTEMP  
IF(NCFIRM.NE.0) WRITE(6,125) INTEMP,UTMCHD  
125 FORMAT(' ZONE. ',12,' (UTM CENTRAL MERIDIAN ',F6.1,' DEG)')  
CALL GETSIN(INTEMP, >1.-1,'EXTRA ZONE SPECIFICATION --')  
IF(OLOCMD.NE.UTMCHD) KSYOWH(WORIG)='NUL' & IF CMD CHANGED MARK ORIG ABSENT  
900 KMD=''  
RETURN  
END
```


DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDGAD
001

SUBROUTINE KMDGAD(8 SADD -- DYNAMIC SADD
U KOMD) 8 1: FIRST 3 CHARS OF COMMAND 0: SPACES

```
C
C
C (E M SCHLOSSER)
C
C
C   INCLUDE KOMXQT.LIST
C   INCLUDE ASMDEF.LIST
C   DIMENSION KSFBUF(8)
C   DATA KSFBUF(8) / ' ' /
C   CALL TRACE
C
C   CALL UNGETS
C   CALL GETSKM(KSFBUF(2),36,-0)
C   IF((FLD(0,30,KSFBUF(2)).NE.'@SADD ').AND.
6   (FLD(0,30,KSFBUF(2)).NE.'@@SADD ')) GO TO 890
C   FLD(0,30,KSFBUF(2))='@ ' 8 BLANK OUT SADD
C   KSFBUF(1)='@START'
C   CALL ERCSF(NA0,KSFBUF)
C   IF(MOD(ASMSB(NA0),5).NE.1) GO TO 810
C   KSFBUF(1)='@SADD.E' 8 BLOCKS USES END-OF-FILE TO DETECT END OF @ADDED ELT
C   MADLEV=MAX0(MADLEV+1,1) 8 INCREMENT NESTED SADD LEVEL
C   CALL ERCSF(NA0,KSFBUF)
C   GO TO 900
C
C
C FLAG BAD SADD ELEMENT
C
C 810 CALL WARNB('BAD SADD FILE OR ELEMENT --')
C   GO TO 900
C
C
C ABNORMAL RETURN
C
C 890 RETURN
C
C
C NORMAL RETURN
C
C 900 KOMD=' '
C   RETURN
C   END
```


DAN PACKAGE APPENDIX M
COMMAND ROUTINES

KND00R
001

SUBROUTINE KND00R(8 SBRKPT -- DYNAMIC SBRKPT
U KOMD) 8 1: FIRST 3 CHARS OF COMMAND 0: SPACES

C
C
C
C
C
C

(E M SCHLOSSER)

INCLUDE KOMXQT.LIST
INCLUDE ASHDEF.LIST
DIMENSION KSFBUF(7)
DATA KSFBUF(7)/* . . .
DIMENSION NULASC(1)
DATA NULASC/0000/
CALL TRACE

C
C
C
C

GET SBRKPT IMAGE

CALL UNGETS
CALL GETSKH(KSFBUF,38,-0)
IF(KSFBUF(1).NE.'SBRKPT') GO TO 800
IF(ASHSI(KSFBUF(2)).NE.'00000 ') GO TO 800
ASHSI(KSFBUF(1))=

C
C
C
C

FIX FOR SYNC PROBLEM IN UNIVAC DEMAND SYMBIONT

DO 500 I=1,6
CALL EAPRNT(0.1,NULASC) 8 PRINT 4 ASCII NULS
CALL ERTWAT(1000) 8 TIMED WAIT 1 SECOND
500 CONTINUE
CALL ERPRCN(1,'R . ') 8 INSERT LOGICAL BREAK INTO PRINTS FILE
CALL ERTWAT(1000) 8 TIMED WAIT 1 SECOND

C
C
C
C

REQUEST SBRKPT

CALL ERCSF(1A0,KSFBUF)
IF(1A0.NE.0) CALL WARN5('BAD SBRKPT --')
KOMD=
RETURN

C
C
C
C

ABNORMAL RETURN

800 RETURN
END

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMDFR
001

SUBROUTINE KMDFR(S SFREE -- DYNAMIC SFREE
U KOMD) S 1: FIRST 3 CHARS OF COMMAND O: SPACES

C
C
C
C
C

(E H SCHLOSSER)

INCLUDE KOMXQT.LIST
CALL TRACE

C

RETURN S NOT YET IMPLEMENTED
END

DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KMDGLO
001

SUBROUTINE KMDGLO(B SLOG -- DYNAMIC SLOG
U KOND) B I: FIRST 3 CHARS OF COMMAND O: SPACES

C
C
C
C
C

(E H SCHLOSSER)

INCLUDE KOKXQT.LIST
INCLUDE ASHDEF.LIST
DIMENSION KSFBUF(13)
DATA KSFBUF(13)/' ' . ''
CALL TRACE

C

CALL UNOETS
CALL GETSKH(KSFBUF,72,-0)
IF(FLD(0,30,KSFBUF(11),NE,'BSLOG ') GO TO 800
ASMSI(KSFBUF(11))=
CALL ERCSF(NAO,KSFBUF)
IF(NAO,NE,0) CALL WARNS('BAD SLOG --')
KOND=
RETURN

C
C
C
C

ABNORMAL RETURN

800 RETURN
END

SUBROUTINE KHXXED: 8 EDIT ACTUAL SPECS INTO MACRO COMMAND DEFINITION

8
I MAKOMD. 8 NAME OF MACRO COMMAND
I NAMFIL. 8 NAME OF FILE
I NAMELT. 8 NAME OF ELEMENT
I LOCFIL) 8 LOCATION WITHIN DSF OF MACRO COMMAND DEFINITION

C

C

C

C HISTORY

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

HISTORY

E M SCHLOSSER LEC 01/17/79 ORIGINAL CODE
E M SCHLOSSER LEC 05/01/79 FIX MISSPELLING OF NQMFL AFTER 130

C METHOD

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C MACHINE-DEPENDENT CODE

UNIVAC EXEC 8 PROGRAM FILE NAMING CONVENTIONS.
DIMENSION 8 FORMAT SPECIFICATIONS ASSUME 8 CHARACTERS PER WORD.

C EXTERNAL REFERENCES

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

SYSADD 8 ADD DISK SYMBOLIC FILE OR ELEMENT TO SYSIN RUNSTREAM
OETS 8 GET REMAINDER OF UNIT 5 BUFFER
OETSKM 8 GET CHARACTER STRING FIELD FROM UNIT 5
WARNS 8 PRINT/LOG WARNING MESSAGE
SYSOET 8 GET NEXT RECORD FROM SYSIN RUNSTREAM
OETCHR 8 GET CHARACTER FROM CHARACTER STRING
PUTCHR 8 PUT CHARACTER INTO CHARACTER STRING
PUTICE 8 PUT I-C-E INTO CHARACTER STRING
KHXXOS 8 GET/EVALUATE ACTUAL SPEC FOR MACRO COMMAND
OETOKM 8 GET CHARACTER STRING DATA FIELD FROM BUFFER
HDWARN 8 SUBMIT WARNING DIAGNOSTIC MESSAGE
HDFATL 8 SUBMIT FATAL DIAGNOSTIC MESSAGE
HOVCST 8 MOVE CHARACTER STRING
NEXTOK 8 GET POINTERS TO NEXT TOKEN
INTEGER LENCST 8 LENGTH OF CHARACTER STRING
LOGICAL TRUCST 8 TRUTH VALUE OF CHARACTER STRING COMPARISON

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C EXCEPTIONS

- C 1. THE FOLLOWING CONDITIONS GENERATE NOTES:
C NO NAME (OR BAD NAME) DECLARED IN MACRO DEFINITION
C
C 2. THE FOLLOWING CONDITIONS GENERATE WARNINGS:
C MACRO COMMAND NESTED MORE THAN 9 DEEP
C MACRO DEFINITION EMPTY
C MACRO DEFINITION LONGER THAN 80 CHARACTERS
C FORMAL SPEC IN MACRO DEFINITION DOES NOT START WITH '<'
C MORE THAN MAXQM FORMAL SPECS DECLARED IN MACRO DEFINITION
C ACTUAL SPEC MISSING
C ACTUAL SPEC STARTS WITH '<'
C ACTUAL SPEC LONGER THAN 12 CHARACTERS
C MORE ACTUAL SPECS THAN FORMAL SPECS
C UNDECLARED SPEC REFERENCED IN MACRO DEFINITION
C MACRO EDIT IMAGE LONGER THAN 80 CHARACTERS
C
C 3. THE FOLLOWING CONDITIONS GENERATE FATAL ERRORS:
C DSE WITH MACRO DEFINITION NOT FOUND
C MACRO DEFINITION CONTAINS EXEC COMMAND
C MACRO EDIT FILE NOT FOUND

C GLOBAL DECLARATIONS
C -----
C

INCLUDE KMXQT.LIST % COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST % DEFINE NULL CHARACTER STRING

C LOCAL DECLARATIONS
C -----
C

PARAMETER MAXQM=% % MAXIMUM NUMBER OF SPEC FIELDS SUPPORTED
INTEGER NQM % SPEC FIELD NUMBER (NUMBER OF QUESTION MARKS)
INTEGER NQMFM % NUMBER OF FORMAL SPEC FIELDS DECLARED
C
INTEGER MAKOMD(3) % ARGUMENT
INTEGER MACDEF(3) % MACRO COMMAND DEFINITION FIELD
INTEGER KHARI % FIRST CHARACTER IN SPEC/TOKEN
INTEGER NDSAVE % VALUE OF NDTOTL ON ENTRY TO SUBROUTINE
C
INTEGER IMGIN(20) % DEFINITION IMAGE INPUT BUFFER
INTEGER LLIN(3) % LOCATION/LENGTH POINTERS FOR IMGIN:
DEFINE LOCIN=LLIN(1) % LOCATION OF PREVIOUS/CURRENT TOKEN IN IMGIN
DEFINE LENIN=LLIN(2) % LENGTH OF PREVIOUS/CURRENT TOKEN IN IMGIN
DEFINE MAXIN=LLIN(3) % LENGTH OF IMGIN
C
INTEGER IMGOUT(14) % EDITED IMAGE OUTPUT BUFFER
INTEGER LLOUT(3) % LOCATION/LENGTH POINTERS FOR IMGOUT:
DEFINE LOCOUT=LLOUT(1) % LOCATION OF PREVIOUS/CURRENT TOKEN IN IMGOUT
DEFINE LENOUT=LLOUT(2) % LENGTH OF PREVIOUS/CURRENT TOKEN IN IMGOUT
DEFINE MAXOUT=LLOUT(3) % LENGTH OF IMGOUT
C
INTEGER % IMOSF(3,MAXQM) % MACHINE DEPENDENT
DEFINE IMOSPF(NQM)=IMOSF(1,NQM) % ACTUAL SPEC FIELD IMAGE BUFFER
INTEGER LENSPP(MAXQM) % TABLE OF LENGTH POINTERS FOR IMOSPF

```

C
      INTEGER INSTAT      & INPUT STATUS
      INTEGER LUEDIT      & LOGICAL UNIT NUMBER FOR EDIT FILE
      INTEGER NFEDIT(2)   & NAME OF EDIT FILE
      INTEGER LOCNXT      & LOCATION TO PUT NEXT CONFIRMATION SPEC FIELD

C
C
C PROCEDURE
C -----
C
C
C      CALL TRACE

C
C
C INITIALIZE
C
      NDSAVE=NDTOTL
      CALL SYSADD(LOCFIL, NAMFIL, NAMELT, ' ')
      IF(LOCFIL.LE.0) CALL MDEFATL( 'NO MACRO DEFINITION IN KMKED')
      IF(MADLEV.GT.9) CALL MARN( 'MACRO COMMAND NESTED TOO DEEP ---')
      LUEDIT=39+MADLEV

C
C
C GET/CHECK NAME FROM MACRO DEFINITION DSF
C
      LOCIN=1      & LOC IS FIRST CHAR OF IMOIN BUFFER
      LENIN=0      & LENGTH OF PREVIOUS FIELD IS 0
      CALL SYSOET(INSTAT,IMOIN,MAXIN,1)      & GET STATUS, IMOIN, LENGTH
      IF(INSTAT.EQ.' ') GO TO 130
      CALL MOWARN( 'MACRO DEFINITION EMPTY')
      GO TO 900
130 MAXIN=LENCST(IMOIN,MAXIN)      & STRIP TRAILING BLANKS, IF ANY
      IF(MAXIN.GT.99) CALL MOWARN( 'MACRO DEFINITION IMAGE TOO LONG')
      MAXIN=MIND(MAXIN,99)+1
      CALL PUTCHR(IMOIN,(MAXIN), ' ')      & APPEND 1 TRAILING BLANK
      MACDEF(1)=' NUL'
      CALL GETOKH(MACDEF,(1),LLIN, LLIN,IMOIN)      & GET NEXT CHAR STR DATA FLD
      IF(TRUCST(MACDEF,1,19, ' ', MAKOMD,1,19)) GO TO 140
      CALL MNOTE( 'NO NAME DECLARED IN MACRO DEFINITION')
      LOCIN=1      & RESTORE LOCLEN POINTER
      LENIN=0
      MAXIN=1+ABS(MAXIN)
      NOMFML=0      & NO FORMAL SPECS ...
      GO TO 200      & ... BUT SEE IF ANY ACTUAL SPECS!!

C
C
C GET/CHECK/TALLY FORMAL SPEC(S) FROM MACRO DEFINITION DSF
C
140 NOMFML=0
      DO 150 NOM=1,99
      MACDEF(1)=' NUL'
      CALL GETOKH(MACDEF,(1),LLIN, LLIN,IMOIN)
      IF(MACDEF(1).EQ.' NUL') GO TO 200      & NO MORE FORMAL SPECS
      NOMFML=MIND(NOMFML+1,MAXOM)
      CALL GETCHR(KHARI, MACDEF,(1))
      IF(KHARI.NE.'(') CALL MOWARN(

```



```

      • 'BAD FORMAL SPEC IN MACRO DEFINITION')
      IF(NQM.GT.MAXQM) CALL MDWARN(
      • 'EXTRA FORMAL SPEC IN MACRO DEFINITION')
180 CONTINUE

C
C GET/CHECK/TALLY ACTUAL SPEC FIELD(S) FROM SYSIN RUNSTREAM
C
200 IF(NQMFML.EQ.0) GO TO 240
   DO 220 NQM=1,NQMFML
      IMOSPF(NQM)=' NUL'
      CALL KMXXQS(IMOSPF(NQM),(18) )
      IF(IMOSPF(NQM).EQ.' NUL') CALL MDWARN( 'MISSING SPEC')
      CALL GETCHR(KHARI, IMOSPF(NQM),(1))
      IF(KHARI.EQ.'<') CALL WARN5( 'BAU SPECIFICATION --')
      LENSPP(NQM)=LENCST(IMOSPF(NQM),18)
      IF(LENSPP(NQM).GT.12) CALL WARN5(
      • 'SPECIFICATION TOO LONG --')
220 CONTINUE

C
C
C FLUSH EXTRA ACTUAL SPECIFICATION(S) FROM SYSIN RUNSTREAM
C
240 ITEMP=' NUL'
   CALL GET5KH(ITEMP,(4), NULCST)
   IF(ITEMP.NE.' NUL') CALL WARN5( 'EXTRA SPECIFICATION --')
   IF(ITEMP.NE.' NUL') GO TO 240

C
C
C IF DIAGNOSTICS ENCOUNTERED FLUSH ADD-ED MACRO DEFINITION
C
   IF(INDSAVE.EQ.NDTOTL) GO TO 500      & NO NEW DIAGNOSTICS?
420 CALL SYSGET(INSTAT,IMGIN,MAXIN )
   IF(INSTAT.EQ.' ') GO TO 420
   IF(INSTAT.NE.'EOA') CALL MDFATL(
   • 'EXEC COMMAND IN MACRO DEFINITION')
   GO TO 900

C
C
C READ NEXT DEFINITION IMAGE
C
500 IF(LOCIN.NE.1) MAXIN=1ABS(MAXIN)      & IF END OF BLOCK, TURN OFF END FLAG
   IF(MAXIN.LE.0) CALL SYSGET(INSTAT,IMGIN,MAXIN )
   IF(INSTAT.NE.' ') GO TO 600      & END OF DEFINITION
   MAXIN=LENCST(IMGIN,MAXIN)      & STRIP TRAILING BLANKS, IF ANY
   IF(MAXIN.GT.80) CALL MDWARN( 'MACRO DEFINITION IMAGE TOO LONG')
   MAXIN=MIND(MAXIN,80)

C
C
C INITIALIZE EDIT IMAGE WITH BLANKS
C
   LOCOUT=1
   CALL MOVCS(IMGOUT,(LOCOUT),(8%), ' ',(11),(11),' ')

C
C
C LOCATE NEXT DEFINITION TOKEN

```

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KHXED
005

```

C
520 CALL NEXTOK(KHARI,LLIN, LLIN,IMGIN, '?', 'NONE')
    IF(MAXIN.LE.0) GO TO 560      @ END OF IMAGE
    IF(KHARI.EQ.'?') GO TO 530

C
C APPEND NON-SPEC TOKEN TO EDIT IMAGE
C
    LENOUT=LENIN
    IF(LENOUT.LE.(81-LOCOUT)) CALL MOVCS(IMGOUT,(LOCOUT),(LENOUT),
        IMGIN,(LOCIN),(LENIN), ' ')
    LOCOUT=LOCOUT+LENOUT
    GO TO 520

C
C APPEND ACTUAL SPEC TO EDIT IMAGE IN PLACE OF FORMAL SPEC TOKEN
C
530 IF(LENIN.GT.NQMFML) CALL MDWARN(
    @ 'UNDECLARED SPEC REFERENCED IN MACRO DEFINITION')
    NQM=MIND(LENIN,NQMFML)
    LENOUT=LENSPF(NQM)
    IF(LENOUT.LE.(81-LOCOUT)) CALL MOVCS(IMGOUT,(LOCOUT),(LENOUT),
        IMGSPF(NQM),(1),(LENOUT), ' ')
    LOCOUT=LOCOUT+LENOUT
    GO TO 520

C
C WRITE EDIT IMAGE
C
560 MAXOUT=LOCOUT-1
    IF(MAXOUT.GT.80) CALL MDWARN( 'MACRO EDIT LINE TOO LONG')
    IF(NDSAVE.EQ.NOTOTL) WRITE(LUEDIT,575) IMGOUT
575 FORMAT(14A6)      @ MACHINE DEPENDENT FORMAT
    GO TO 500

C
C DRAIN READS BUFFER & APPEND ITS CONTENTS (IF ANY) TO EDIT FILE
C
600 IMGOUT(1)= ' NUL'
    CALL GETS(IMGOUT,(84), NULCST)
    IF(IMGOUT(1).NE.' NUL') WRITE(LUEDIT,615) IMGOUT
615 FORMAT(14A6)

C
C CLOSE EDIT FILE AND ADD TO SYSIN RUNSTREAM
C
    IF(INSTAT.NE.'EOA') CALL MDFATL(
    @ 'EXEC COMMAND IN MACRO DEFINITION')
    ENOFIL LUEDIT
    CALL CLOSE( LUEDIT,1)      @ CLOSE & REWIND
    IF(NDSAVE.NE.NOTOTL) GO TO 900
    NFEDIT(1)= ' '
    NFEDIT(2)= ' '
    CALL PUTICE(NFEDIT,(1), ICE('0')+LUEDIT/10)      @ TENS DIGIT
    CALL PUTICE(NFEDIT,(2), ICE('0')+LUEDIT-10*(LUEDIT/10)) @ UNITS DIGIT
    CALL SYSADD(LOCFIL, NFEDIT, ' ', ' ')

```

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMXXED
006

```
      IF(LOCFIL.LT.0) CALL MDFATL( 'NO MACRO EDIT FILE IN KMXXED')
C
C
C CONFIRM MACRO COMMAND & SPECS
C
      IF(MCFIRM.EQ.0) GO TO 900
      CALL MOV CST(IMGIN.(1).(120).
        *      MAKOMD.(1).(12). ' ')
      LOCNXT=LENCST(IMGIN.120)+1
      IF(NQMFML.LE.0) GO TO 730
      DO 710 NQM=1,NQMFML
        IF(LOCNXT.GT.105) GO TO 730
        CALL PUTCHR(IMGIN.(LOCNXT). ' ')
        LOCNXT=LOCNXT+2
        CALL MOV CST(IMGIN.(LOCNXT).(12).
          *      IMOSPF(NQM).(1).(12). ' ')
        LOCNXT=LOCNXT+LENSPF(NQM)
710      CONTINUE
730 WRITE(6.735) IMGIN
735 FORMAT(1X.20A6)
C
C
C DONE
C
900 RETURN
      END
```

SUBROUTINE KHXXOS(: GET/EVALUATE ACTUAL SPEC FOR MACRO COMMAND
0 KHFLD. : ACTUAL SPEC FIELD (UNCHANGED IF MISSING)
(KHLEN : LENGTH IN CHARS (WILL BE PADDED WITH BLANKS TO WORD PAD)
")

C
C
C
C HISTORY
C -----

C E H SCHLOSSER LEC 01/17/79 ORIGINAL CODE
C
C

C
C METHOD
C -----

C GET ACTUAL SPECIFICATION (IF ANY). IF ITS AN EXPRESSION (ENCLOSED
C IN PARENTHESES) CHECK IT. EVALUATE IT. AND RETURN ITS VALUE.
C
C

C
C MACHINE-DEPENDENT CODE
C -----

C NONE.
C
C

C
C EXTERNAL REFERENCES
C -----

C GETSKH : GET CHARACTER STRING FIELD FROM UNIT 5
C MOVCS : MOVE CHARACTER STRING
C INTEGER ICE : INTEGER-CHARACTER-EQUIVALENT
C INTEGER LCHREQ : LOCATE CHARACTER IN STRING EQUAL TO SEARCH CHARACTER
C INTEGER LENPAD : LENGTH IN CHARACTERS INCL PAD TO WORD BOUNDARY
C
C

C
C EXCEPTIONS
C -----

C 1. AN INVALID EXPRESSION IS RETURNED WITH THE LEADING '(' CHANGED
C TO '<'.
C
C

C
C GLOBAL DECLARATIONS
C -----

C INCLUDE KOMXQT.LIST : COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C INCLUDE KOMKLS.LIST : COMMON CLASSIFICATION INFO
C INCLUDE NULCST.LIST : DEFINE NULL CHARACTER STRING
C
C

C
C LOCAL DECLARATIONS
C -----

C INTEGER KHXP : EXPRESSION
C INTEGER MODSET : MODE SETTING OF SOME MODE SWITCH
C

DAM PACKAGE APPENDIX H
COMMAND ROUTINES

KHXXOS
002

```
C
C
C PROCEDURE
C -----
C
C
C CALL TRACE
C
C
C GET ACTUAL SPECIFICATION FIELD
C
C CALL GETSKH(KHFLD.(KHLEN), NULCST)
C IF(ICE(KHFLD).NE.ICE('')) GO TO 900      & NOT AN EXPRESSION
C
C
C EXTRACT 1ST 3 CHARS OF EXPRESSION FROM ENCLOSING PARENTHESES
C
C IF(ILCHREQ(KHFLD.(2).(3).''').NE.0) GO TO 240
C CALL PUTCHR(KHFLD.(1). '<')      & FLAG AS INVALID EXPRESSION
C GO TO 900
C 240 CALL MOVCST(KHEXP.(1).(LENPAD(3)).
C      '      KHFLD.(2).(3).'' ')
C
C
C CHECK/EVALUATE GENERAL EXPRESSION
C
C IF(KHEXP.NE.'DET') GO TO 310      & DETECT?
C CALL MOVCST(KHFLD.(1).(LENPAD(KHLEN)).
C      '      KLSTYP.(1).(3).'' ')
C GO TO 900
C 310 CONTINUE      & FUTURE CODE
C
C
C CHECK/EVALUATE MODE SWITCH EXPRESSION
C
C 500 IF(KHEXP.NE.'BAT') GO TO 510      & BAT(CH)?
C      MODSET=MBATCH
C      GO TO 700
C 510 IF(KHEXP.NE.'CHE') GO TO 520      & CHECKOUT?
C      MODSET=MCHECK
C      GO TO 700
C 520 IF(KHEXP.NE.'CON') GO TO 530      & CONFIRM?
C      MODSET=MCFIRM
C      GO TO 700
C 530 IF(KHEXP.NE.'DUM') GO TO 540      & DUM(P)?
C      MODSET=MDUMP
C      GO TO 700
C 540 IF(KHEXP.NE.'ECH') GO TO 550      & ECH(O)?
C      MODSET=MECHO
C      GO TO 700
C 550 IF(KHEXP.NE.'LEG') GO TO 560      & LEG(EN)?
C      MODSET=MLEOND
C      GO TO 700
C 560 IF(KHEXP.NE.'PRO') GO TO 570      & PROMPT?
C      MODSET=MPROMT
C      GO TO 700
```

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

KMXXOS
003

```
570 IF(KHEXP.NE.'TRA') GO TO 580      & TRAI1?
      MODSET=MTRACE
      GO TO 700
580 CONTINUE      & FUTURE CODE
C
C
C INVALID EXPRESSION -- PUT '<' IN FIRST CHARACTER
C
      CALL PUTCHR(KHFLD.(1), '<')
      GO TO 900
C
C
C STORE EVALUATION OF MODE SWITCH EXPRESSION
C
700 KHEXP='ON'
      IF(MODSET.EQ.0) KHEXP='OFF'
      CALL MOVCST(KHFLD.(1),(LENPAD(KHLEN)),
      *           KHEXP.(1),(3),' ')
C
C
C DONE
C
900 RETURN
      END
```

DAM PACKAGE APPENDIX N
UTILITY ROUTINES

APPENDIX-N
001

BPRT.SC DAM.PREFACE-N	. (8009)	SET TABS 8 12 & 31
BPRT.SC DAM.APPENDIX-N	.	
BPRT.SC DAM.ARE077	. DUMP A-REGISTER IN OCTAL (FOR DEBUGGING)	
BPRT.SC DAM.ARGRET	. DETERMINE # OF ACTUAL ARGUMENTS & RETURN VECTOR	
BPRT.SC DAM.ATRACE/DAM	. TRACE CALLS TO ASSEMBLER ROUTINES	
BPRT.SC DAM.BOX-CHR	. BOX CHARACTERS	
BPRT.SC DAM.BYTDMP	. EXTRACT/SCALE/DUMP BYTE FIELDS (2'S COMPL)	
BPRT.SC DAM.CALCHA	. CALIBRATE CHANNEL POINTERS	
BPRT.SC DAM.CALCOL	. CALIBRATE COLOR/INTENSITY IN SYMBOL TABLE	
BPRT.SC DAM.CALSCA	. CALIBRATE PRINT/PLOT COEFFICIENTS FOR SCALE	
BPRT.SC DAM.CALSPA	. CALIBRATE PRINT/PLOT COEFFICIENTS FOR SPACING	
BPRT.SC DAM.CALSYM	. CALIBRATE SYMBOL TABLE FOR PRINTING	
BPRT.SC DAM.CALWIN	. CALIBRATE WINDOW ENVELOPES	
BPRT.SC DAM.CLOSE3	. CLOSE UNIT 3 (INPUT ERTS MSS DATA)	
BPRT.SC DAM.CLOSE4	. CLOSE COMMAND RECALL FILE (UNIT 4)	
BPRT.SC DAM.CLOSPR	. CLOSE AND PRINT ALTERNATE PRINT (SPOOL) FILES	
BPRT.SC DAM.CLSHDG	. WRITE COMMON ID/CLASSIFICATION HEADING	
BPRT.SC DAM.CL3BIP	. CLOSE UNIT 3 (INPUT ERTS MSS DATA IN BIP FORMAT)	
BPRT.SC DAM.CORLT	. CORRELATIONS/MEANS/DEVS FROM SUMS/SUMS-OF-PRODS	
BPRT.SC DAM.CROPOW	. CROP MSS OUTPUT WINDOW TO FIT PRINT FILE SIZE	
BPRT.SC DAM.DCORLT	. CORRELATIONS/MEANS/DEVS FM D P SUMS/SUMS-OF-PROD	
BPRT.SC DAM.DEG	. CONVERT DEGREES.MINUTES.SECONDS TO DEGREES	
BPRT.SC DAM.DELETE-DENS	. DELETE ALL DENSITY FILES	
BPRT.SC DAM.DDJR/MATHPACK	. DBL PRECISION GAUSS-JORDAN REDUCTION	
BPRT.SC DAM.DGPCNT	. COMPUTE % OF TRACE FOR MATRIX DIAGONAL ELEMENTS	
BPRT.SC DAM.DGRECP	. RECIPROCAL OF MATRIX DIAGONAL ELEMENTS	
BPRT.SC DAM.DQSQT	. SQUARE ROOT OF MATRIX DIAGONAL ELEMENTS	
BPRT.N .DLETFR	. DELETE ALTERNATE PRINT FILES (SEE CLOSPR)	
BPRT.N .DLET2N	. DELETE INPUT DETECTION FILES (SEE OPN12N)	
BPRT.SC DAM.DMPTIC	. DUMP TICK TABLE	
BPRT.SC DAM.DMPWIN	. DUMP WINDOW PACKET	
BPRT.SC DAM.DSSPR	. COMPUTE DBL PRECISION SUMS & SUMS-OF-PRODUCTS	
BPRT.SC DAM.DSSPR3	. DBL PREC SUMS & SUMS-OF-PRODUCTS FROM UNIT 3	
BPRT.SC DAM.D2DMS	. DEGREES TO DEGREES. MINUTES. SECONDS	
BPRT.SC DAM.EISRTO	. SORT E-VALUES/E-VECTORS BY DESCENDING E-VALUES	
BPRT.SC DAM.ENVORI	. ADD ORIGIN TO ENVELOPE (REAL WINDOW PACKET)	
BPRT.SC DAM.ENVWIN	. COMPUTE ENVELOPE FOR REAL WINDOW PACKET	
BPRT.SC DAM.EOF	. END-OF-FILE (EOF)	
BPRT.SC DAM.FACPRT	. PRINT FACTOR STRUCTURE/COEFFICIENTS/MEANS	
BPRT.SC DAM.FLINFO	. GET FILE DESCRIPTIVE INFORMATION	
BPRT.SC DAM.GCERT	. LOAD 'ERT' GEOMETRIC CONSTANTS	
BPRT.SC DAM.GCHOM	. LOAD 'HOM' GEOMETRIC CONSTANTS	
BPRT.N .OCLCC	. LOAD 'LCC' GEOMETRIC CONSTANTS (SEE GCHOM)	
BPRT.SC DAM.GCONST	. LOAD PROPER GEOMETRIC CONSTANTS	
BPRT.N .GCPS	. LOAD 'PS' GEOMETRIC CONSTANTS (SEE GCHOM)	
BPRT.N .GCSON	. LOAD 'SON' GEOMETRIC CONSTANTS (SEE GCHOM)	
BPRT.N .GCUTH	. LOAD 'UTH' GEOMETRIC CONSTANTS (SEE GCHOM)	
BPRT.SC DAM.GENTIC	. GENERATE/STORE/LIST/SORT TICKS	
BPRT.SC DAM.GETDSR	. GET DISK SYMBOLIC RECORD	
BPRT.SC DAM.GETRAD	. GET ORIGINAL/TRANSFORMED RADIANCE FROM ERTS TAPE	
BPRT.SC DAM.GETS	. GET REMAINING CONTENTS OF UNIT 5 BUFFER	
BPRT.N .GET5AL	. GET ALPHA DATA FIELD FROM UNIT 5 (SEE GETS)	
BPRT.N .GET5FR	. GET FRACTION DATA FIELD FROM UNIT 5 (SEE GETS)	
BPRT.N .GET5IN	. GET INTEGER DATA FIELD FROM UNIT 5 (SEE GETS)	
BPRT.N .GET5KH	. GET CHARACTER DATA FIELD FROM UNIT 5 (SEE GETS)	
BPRT.N .GET5RL	. GET REAL DATA FIELD FROM UNIT 5 (SEE GETS)	

AMS0.N	.GET9SX	. GET SEX'Y DATA FIELD FROM UNIT 5 (SEE GETS)
SPRT.SC	DAM.HDUNIT	. PRINT HEADING LINE(S)
SPRT.SC	DAM.1B4350	. ALLOCATE ARRAY OF 4350 WORDS IN 1-BANK
SPRT.SC	DAM.IDERT	. PRINT SHORT ERTS SCENE IDENTIFICATION
SPRT.SC	DAM.IDERTS	. PRINT COMPLETE ERTS SCENE IDENTIFICATION
SPRT.SC	DAM.IDLU3	. PRINT SHORT ID FOR LOGICAL UNIT 3
SPRT.SC	DAM.IDUP	. INTEGER DUPLICATE (INDIRECT REF TO OPTIONAL ARG)
AMS0.N	.INSTAT	. GET FORTRAN I/O STATUS (UNIVAL SYSTEM ROUTINE)
SPRT.SC	DAM.INVORI	. ADD ORIGIN TO ENVELOPE (INTEGER WINDOW PACKET)
SPRT.SC	DAM.INVWIN	. COMPUTE ENVELOPE FOR INTEGER WINDOW PACKET
SPRT.SC	DAM.14KOLR	. INTEGER-COLOR-EQUIVALENT FOR COLOR
SPRT.SC	DAM.JACHX/MATHPACK	. JACOBI ITERATION TO FIND EIGEN-VALUES/VECTORS
SPRT.SC	DAM.JOIN2N	. JOIN BUFFERS FROM TWO DETECTION FILES
SPRT.SC	DAM.KOLR4I	. COLOR FOR INTEGER-COLOR-EQUIVALENT
SPRT.SC	DAM.KSPRED	. SPREAD COUNT FLAGS INTO INTERIOR UNDEFINED PIXLS
SPRT.SC	DAM.LBOX4I	. LINE OF BOX DIGIT FOR INTEGER
AMS0.N	.LDREON	. LOAD NOMINAL REGISTRATION PARAMETERS (SEE LDREOB)
SPRT.SC	DAM.LDREOB	. LOAD EXACT REGISTRATION PARAMETERS FROM UNIT 8
SPRT.SC	DAM.LOCDSF	. LOCATE DISK SYMBOLIC FILE OR ELEMENT
SPRT.SC	DAM.LOG2	. LOGARITHM, BASE 2 (TRUNCATED) OF INTEGER
SPRT.SC	DAM.MAPHDG	. WRITE MAP WINDOW HEADING
SPRT.SC	DAM.MATPRT	. PRINT MATRIX
AMS0.N	.MOCLRF	. CLEAR 'FATAL ERROR' COUNT (SEE MLOG)
AMS0.N	.MOCLRW	. CLEAR 'WARNING' COUNT (SEE MLOG)
AMS0.N	.MOFATL	. PRINT/LOG/COUNT 'FATAL ERROR' (SEE MLOG)
SPRT.SC	DAM.MOLOG	. LOG DIAGNOSTIC MESSAGES
AMS0.N	.MONOTE	. PRINT/LOG 'NOTE' (SEE MLOG)
AMS0.N	.MOWARN	. PRINT/LOG/COUNT 'WARNING' (SEE MLOG)
SPRT.SC	DAM.MSKPIX	. MASK PIXELS IN BUFFER OUTSIDE NON-TRIVIAL WINDOW
SPRT.SC	DAM.MVCONT	. MOVE CONTENTS BETWEEN SPECIFIED LOCATIONS
SPRT.SC	DAM.MXMLT/MATHPACK	. MATRIX MULTIPLICATION
SPRT.SC	DAM.NEGPIC	. CONVERT DIGITAL PICTURE FROM POSITIVE TO NEGATIVE
SPRT.SC	DAM.NTABS/DAM	. I/O UNIT NUMBER TABLE
SPRT.SC	DAM.NULSUB	. DO ABSOLUTELY NOTHING!!
SPRT.SC	DAM.NVIATO	. NAME 'VIA' 'TO' ROUTINES
SPRT.SC	DAM.OPENPR	. OPEN ALTERNATE PRINT (SPOOL) FILES (UNITS 10-19)
SPRT.SC	DAM.OPEN3	. OPEN UNIT 3 (INPUT ERTS MSS DATA)
AMS0.N	.OPEN4	. OPEN COMMAND RECALL FILE (SEE WRITE4)
SPRT.SC	DAM.OPN12N	. OPEN INPUT DETECTION FILE(S) (UNITS 21-24)
SPRT.SC	DAM.OP3BIP	. OPEN UNIT 3 (INPUT ERTS MSS DATA IN BIP FORMAT)
SPRT.SC	DAM.OP3DSK	. OPEN UNIT 3 (INPUT DATA ON DISK IN PXBDEF FMT)
SPRT.SC	DAM.OP3MDP	. OPEN UNIT 3 (INPUT ERTS MSS DATA IN MDP FORMAT)
SPRT.SC	DAM.OP3TAP	. OPEN UNIT 3 (INPUT ERTS MSS DATA ON TAPE)
SPRT.SC	DAM.O3ANCL	. OPEN UNIT 3 (MDP FORMAT ANCILLARY RECORDS)
SPRT.SC	DAM.O3ANOT	. OPEN UNIT 3 (MDP FORMAT ANNOTATION RECORDS)
SPRT.SC	DAM.O3HDR	. OPEN UNIT 3 (MDP FORMAT HEADER RECORD)
SPRT.SC	DAM.O3SZAM	. OPEN UNIT 3 (MDP AM FORMAT: SIZE & INPUT WINDOW)
SPRT.SC	DAM.O3SZAR	. OPEN UNIT 3 (MDP AR FORMAT: SIZE & INPUT WINDOW)
SPRT.SC	DAM.O3SZPH	. OPEN UNIT 3 (MDP PH FORMAT: SIZE & INPUT WINDOW)
SPRT.SC	DAM.O3SZPR	. OPEN UNIT 3 (MDP PR FORMAT: SIZE & INPUT WINDOW)
SPRT.SC	DAM.O3TOR	. OPEN UNIT 3 (MDP FORMAT TAPE DIRECTORY RECORD)
AMS0.N	.PABORT	. PROGRAM ABORT (SEE PSTOP)
SPRT.SC	DAM.PITROL	. ESTIMATE PITCH AND ROLL
SPRT.SC	DAM.PRNUM	. PRINT BOX NUMBERS (VARIABLE HEIGHT)
SPRT.SC	DAM.PROVF1	. PRINT/OVERPRINT FILES
SPRT.SC	DAM.PROVSY	. PRINT/OVERPRINT SYMBOL BUFFER

APRT.SC DAM.PRSYML	. PRINT SYMBOL LEGEND
APRT.SC DAM.PRTCHR	. WRITE BOX CHARACTERS ON ANY UNIT
APRT.SC DAM.PRTINC	. SET PRINT LINES PER INCH
APRT.SC DAM.PRTHRO	. SET PRINT MARGINS & LINES PER PAGE
APRT.SC DAM.PSTART	. PROGRAM INITIATION
APRT.SC DAM.PSTOP	. PROGRAM TERMINATION
APRT.SC DAM.PX80MP	. DUMP PXBDEF PIXEL BUFFER PREAMBLE (FOR DEBUGGING)
APRT.SC DAM.PX4AM	. PXBDEF PREAMBLE FOR MDP 'AM' PREAMBLE
APRT.SC DAM.PX4AR	. PXBDEF PREAMBLE FOR MDP 'AR' PREAMBLE
APRT.SC DAM.PX4PM	. PXBDEF PREAMBLE FOR MDP 'PM' PREAMBLE
APRT.SC DAM.PX4PR	. PXBDEF PREAMBLE FOR MDP 'PR' PREAMBLE
APRT.SC DAM.QUAD	. FIT $Y = A \cdot X^{+2} + B \cdot X + C$ & SOLVE FOR EXTREMUM
APRT.SC DAM.QUARTN	. NORMALIZED QUARTIMAX ROTATION CRITERION
APRT.SC DAM.QUARTU	. UN-NORMALIZED QUARTIMAX ROTATION CRITERION
APRT.SC DAM.RD3BIL	. READ UNIT 3 (ERTS MSS DATA IN MDP BIL FORMAT)
APRT.SC DAM.RD3BIP	. READ UNIT 3 (ERTS MSS DATA IN BIP FORMAT)
APRT.SC DAM.RD3BSQ	. READ UNIT 3 (ERTS MSS DATA IN MDP BSQ FORMAT)
APRT.SC DAM.RD3DSK	. READ UNIT 3 (DATA ON DISK IN PXBDEF FORMAT)
APRT.SC DAM.RD3NUL	. READ UNIT 3 (SYNTHETIC DATA WHEN NO UNIT 3)
APRT.SC DAM.READ2N	. READ DATA FROM DETECTION FILE(S) (UNITS 21-24)
APRT.SC DAM.READ3	. READ UNIT 3 (ERTS MSS DATA)
APRT.SC DAM.READ5	. FILL BUFFER FOR UNIT 5 (CARD READER OR TERMINAL)
APRT.SC DAM.REG-NOM/LSAT-1-ERT	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REG-NOM/LSAT-2-ERT	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REG-NOM/LSAT-2-HOM	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REG-NOM/LSAT-2-SOM	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REG-NOM/LSAT-2-UTH	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REG-NOM/LSAT-3-ERT	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REG-NOM/LSAT-3-HOM	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REG-NOM/LSAT-3-SOM	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REG-NOM/LSAT-3-UTH	. NOMINAL REGISTRATION PARAMETERS
APRT.SC DAM.REVERT	. REVERT EQUATIONS
APRT.SC DAM.RITADD	. WRITE/ADD SPECIFIED SYMBOLIC ELEMENT
APRT.SC DAM.RL2ISX	. CONVERT REAL TO SEXAGENARY ARRAY (INTEGER)
APRT.SC DAM.RL2SX	. CONVERT REAL TO SEXAGENARY ARRAY (REAL)
APRT.SC DAM.RL4SX	. COMPUTE REAL FROM SEXAGENARY ARRAY (REAL)
APRT.SC DAM.ROTCHX	. ROTATE TWO MATRIX COLUMNS TO MAXIMIZE FUNCTION
APRT.SC DAM.ROTCOL	. ROTATE TWO MATRIX COLUMNS
APRT.SC DAM.ROTROW	. ROTATE TWO MATRIX ROWS
APRT.SC DAM.R3TREC	. READ UNIT 3 (READ ONE RECORD FROM TAPE)
APRT.SC DAM.SETMOD	. GET/SET MODE SWITCHES
APRT.SC DAM.SHASAM	. SHARPEN SAMPLES IN PXBDEF FORMAT BUFFER
APRT.SC DAM.SHFTBC	. SHIFT BITS CIRCULAR WITHIN WORDS OF ARRAY
APRT.SC DAM.SPANS	. ENABLE/DISABLE SPANNING FOR UNIT 5
APRT.SC DAM.SPLIT	. SPLIT REAL INTO SIGN, INTEGER, DECIMAL
APRT.SC DAM.SREAD5	. SPANNED READ OF UNIT 5 (USED ONLY BY GET5)
APRT.SC DAM.SSPR	. COMPUTE SUMS & SUMS-OF-PRODUCTS
APRT.SC DAM.STREG8	. STORE REGISTRATION PARAMETERS ON UNIT 8
APRT.SC DAM.SUBWIN	. GENERATE SUBWINDOW MAPS
APRT.SC DAM.TRACE/DAM	. TRACE CALLS TO FORTRAN ROUTINES
APRT.SC DAM.TRECVR	. TAPE ERROR RECOVERY FOR UNIT 3 (MDP FORMAT TAPES)
APRT.SC DAM.TSWAP3	. TAPE SWAP FOR UNIT 3 (MDP FORMAT TAPES)
ANSO.N UNGET5	. BACK UP 1 DATA FIELD ON UNIT 5 INPUT (SEE GET5)
APRT.SC DAM.VALKEY	. VALIDATE SECURITY KEY
APRT.SC DAM.VAR3QN	. NORMALIZED VARIMAX ROTATION CRITERION
APRT.SC DAM.VARSQU	. UN-NORMALIZED VARIMAX ROTATION CRITERION

DAM PACKAGE APPENDIX N
UTILITY ROUTINES

APPENDIX-N
004

SPRT.SC DAM.VERA40
SPRT.SC DAM.VERA4P
SPRT.SC DAM.VERO4U
SMSO.N .VIATO
SPRT.SC DAM.WARNS
SPRT.SC DAM.WINEXT
SPRT.SC DAM.WININT
SPRT.SC DAM.WRITE4
SPRT.SC DAM.WRVERT
SPRT.SC DAM.XREG77

- . WINDOW VERTICES: ADJUSTED MSS FOR GEOGRAPHIC
- . WINDOW VERTICES: ADJUSTED MSS FOR PRINT/PLOT
- . WINDOW VERTICES: GEOGRAPHIC FOR UTM
- . CALL 'VIA' 'TO' ROUTINES (SEE NVIATO)
- . PROCESS WARNING DIAGNOSTIC FOR UNIT 5
- . COMPUTE INTERCEPTS FOR WINDOW EXTERIOR
- . COMPUTE INTERCEPTS FOR WINDOW INTERIOR
- . WRITE COMMAND RECALL FILE (UNIT 4)
- . WRITE VERTEX COORDINATES FOR WINDOW
- . DUMP X-REGISTER IN OCTAL (FOR DEBUGGING)

DAM PACKAGE APPENDIX O
COORDINATE TRANSFORMATIONS

APPENDIX-O
001

SPRT.SC DAM.PREFACE-O	. (0009)	SET TABS 8 31
SPRT.SC DAM.APPENDIX-O	.	
SPRT.SC DAM.A40	. ADJUSTED MSS COORDINATES FOR GEOGRAPHIC COORD	
SPRT.SC DAM.A4P	. ADJUSTED MSS COORDINATES FOR PRT/PLT COORDINATES	
SPRT.SC DAM.DU40/CLARKE1866	. DBL PRECISION UTM COORDINATES FOR GEOGRAPHIC	
SPRT.SC DAM.04A	. GEOGRAPHIC COORDINATES FOR ADJUSTED MSS COORD	
SPRT.SC DAM.04P	. GEOGRAPHIC COORDINATES FOR PRT/PLT COORD	
SPRT.SC DAM.04U/CLARKE1866	. GEOGRAPHIC COORDINATES FOR UTM COORDINATES	
SPRT.SC DAM.P4A	. PRT/PLT COORDINATES FOR ADJUSTED MSS COORDINATES	
SPRT.SC DAM.P40	. PRT/PLT COORDINATES FOR GEOGRAPHIC COORDINATES	
SPRT.SC DAM.U40/CLARKE1866	. UTM COORDINATES FOR GEOGRAPHIC COORDINATES	

DAM PACKAGE APPENDIX P
EXECUTIVE REQUESTS

APPENDIX-P
001

SPRT.SC	DAM.PREFACE-P	. (0009)	SET TABS 8 12 6 31
SPRT.SC	DAM.APPENDIX-P	.	
SPRT.SC	DAM.EAPRNT/DAM	. PRINT IMAGE ON TTY OR LINE PRINTER (ASCII)	
SPRT.SC	DAM.EAREAD/DAM	. READ IMAGE FROM TTY OR CARD READER (ASCII)	
SPRT.SC	DAM.ERCFS/DAM	. SUBMIT EXEC COMMANDS (FIELDATA)	
SPRT.SC	DAM.ERDATE/DAM	. RETURN SYSTEM DATE AND TIME (FIELDATA)	
SPRT.SC	DAM.ERERR/DAM	. ERRS TERMINATE PROGRAM	
SPRT.SC	DAM.EREXIT/DAM	. TERMINATE PROGRAM IMMEDIATELY	
SPRT.SC	DAM.ERFACL/DAM	. RETRIEVE FACILITIES ASSIGNMENT INFORMATION	
SPRT.SC	DAM.ERFITH/DAM	. RETRIEVE FACILITIES ASSIGNMENT INFORMATION	
SPRT.SC	DAM.ERINFO/DAM	. RETRIEVE SYSTEM/RUN/PROGRAM/FILE INFO	
SPRT.SC	DAM.ERIO/DAM	. INITIATE I/O	
SPRT.SC	DAM.ERIOH/DAM	. INITIATE I/O & WAIT FOR COMPLETION	
SPRT.SC	DAM.ERPCHA/DAM	. WRITE IMAGE TO ALTERNATE PUNCH FILE (FIELDATA)	
SPRT.SC	DAM.ERPCT/DAM	. RETRIEVE PART OF PROGRAM CONTROL TABLE (PCT)	
SPRT.SC	DAM.ERPFS/DAM	. PROGRAM FILE SEARCH FOR INFO ON ELEMENT	
SPRT.SC	DAM.ERPRCA/DAM	. SET ALTERNATE PRINT FILE CONTROLS (FIELDATA)	
SPRT.SC	DAM.ERPRCN/DAM	. SET PRINT FILE CONTROLS (FIELDATA)	
SPRT.SC	DAM.ERPRNT/DAM	. PRINT IMAGE ON TTY OR LINE PRINTER (FIELDATA)	
SPRT.SC	DAM.ERPRTA/DAM	. WRITE IMAGE TO ALTERNATE PRINT FILE (FIELDATA)	
SPRT.SC	DAM.ERREAD/DAM	. READ IMAGE FROM TTY OR CARD READER (FIELDATA)	
SPRT.SC	DAM.ERREDA/DAM	. READ IMAGE FROM READ ALT FILE (FIELDATA)	
SPRT.SC	DAM.ERSUPS/DAM	. RETURN ACCUMULATED SUPS (200 USEC INCR) FROM PCT	
BNSO.N	.ERTRAN	. SUBMIT EXEC REQUESTS (UNIVAC SYSTEM ROUTINE)	
SPRT.SC	DAM.ERTSWP/DAM	. SWAP TAPE REELS OF MULTI-REEL FILE	
SPRT.SC	DAM.ERTWAT/DAM	. TIMED WAIT UP TO 30 SECONDS	
SPRT.SC	DAM.ERWAIT/DAM	. WAIT FOR COMPLETION OF I/O	
BNSO.N	.NERTRN	. SUBMIT EXEC REQUESTS (UNIVAC SYSTEM ROUTINE)	
SPRT.SC	DAM.SYPOET/DAM	. OUTPUT PROMPT RECORD & GET NEXT SYSIN RECORD	
SPRT.SC	DAM.SYSADD/DAM	. ADD DISK SYMBOLIC FILE OR ELT TO SYSIN RUNSTREAM	
SPRT.SC	DAM.SYSOET/DAM	. GET NEXT RECORD FROM SYSIN RUNSTREAM	
SPRT.SC	DAM.SYSPUT/DAM	. OUTPUT RECORD TO SYSOUT PRIMARY CRT/PRINTED FILE	

SPRT.SC	DAH.PREFACE-Q	. (7912)	SET TABS 8 12 & 31
SPRT.SC	DAH.APPENDIX-Q	.	
SN50.N	.ADJL4C	. ADJUSTED LINE FOR CORRECTED (SEE TRFORM-PROCS)	
SN50.N	.ADJS4C	. ADJUSTED SAMPLE FOR CORRECTED (SEE TRFORM-PROCS)	
SPRT.SC	DAH.ALTPRT-PROCS	. COMMON/DEFINE FOR ALTERNATE PRINT FILES	
SPRT.SC	DAH.ASMDEF-PROC	. DEFINE UNIVAC ASSEMBLER PARTIAL WDS IN FORTRAN V	
SN50.N	.ASMM1 ... ASMM2	. PARTIAL HALF-WORD MNEMONICS (SEE ASHDEF-PROC)	
SN50.N	.ASHS1 ... ASHLS	. PARTIAL SIXTH-WORD MNEMONICS (SEE ASHDEF-PROC)	
SN50.N	.ASMT1 ... ASMT3	. PARTIAL THIRD-WORD MNEMONICS (SEE ASHDEF-PROC)	
SN50.N	.AXRS	. STANDARD 1100 REG MNEMONICS (UNIVAC SYSTEM PROC)	
SPRT.SC	DAH.CBDEF-PROC	. DEFINE CHARACTER BUFFER STRUCTURE & STD CB'S	
SN50.N	.CORL4A	. CORRECTED LINE FOR ADJUSTED (SEE TRFORM-PROCS)	
SN50.N	.COR54A	. CORRECTED SAMPLE FOR ADJUSTED (SEE TRFORM-PROCS)	
SN50.N	.CORL4P	. CORRECTED LINE FOR PPD (SEE TRFORM-PROCS)	
SN50.N	.COR54P	. CORRECTED SAMPLE FOR PPD (SEE TRFORM-PROCS)	
SN50.N	.CORL4S	. CORRECTED LINE FOR STM (SEE TRFORM-PROCS)	
SN50.N	.COR54S	. CORRECTED SAMPLE FOR STM (SEE TRFORM-PROCS)	
SN50.N	.DIOITS	. (SEE FORPROCS)	
SPRT.SC	DAH.FACBIT-PROC	. MNEMONICS FOR EXEC-8 CSFS FACILITY STATUS BITS	
SPRT.SC	DAH.FIDEF-PROC	. DEFINE STRUCTURE OF FILE DEFINITION RECORD	
SN50.N	.FLDEF	. (SEE FORPROCS)	
SPRT.SC	DAH.FORPROCS	. MISCELLANEOUS DEFINE PROCEDURES	
SPRT.SC	DAH.GETOPT-APROC	. ASSEMBLER MANIPULATION OF XQT OPTION BITS/LETTER	
SN50.N	.ICBUF1	. STANDARD CHARACTER BUFFER #1 (SEE CBDEF-PROC)	
SN50.N	.ICBUF2	. STANDARD CHARACTER BUFFER #2 (SEE CBDEF-PROC)	
SN50.N	.ICBUF3	. STANDARD CHARACTER BUFFER #3 (SEE CBDEF-PROC)	
SN50.N	.ICBUF4	. STANDARD CHARACTER BUFFER #4 (SEE CBDEF-PROC)	
SN50.N	.IOADDR	. I/O ADDRESS OF BUFFER (SEE KOMIO-PROC)	
SN50.N	.IOAFCT	. I/O ABNORMAL FRAME COUNT (SEE KOMIO-PROC)	
SN50.N	.IOCODE	. I/O STATUS CODE MNEMONIC (SEE KOMIO-PROC)	
SN50.N	.IOFUNC	. I/O FUNCTION (SEE KOMIO-PROC)	
SN50.N	.IONMDS	. I/O NUMBER OF WDS TRANSMITTED (SEE KOMIO-PROC)	
SN50.N	.IOSECT	. I/O SECTOR IN FILE (SEE KOMIO-PROC)	
SN50.N	.IOSIZE	. I/O BUFFER SIZE (SEE KOMIO-PROC)	
SN50.N	.IOSTAT	. I/O STATUS NUMBER (SEE KOMIO-PROC)	
SN50.N	.IOWAIT	. I/O WAIT SPEC (SEE KOMIO-PROC)	
SN50.N	.IOWORD	. I/O WORD IN FILE (SEE KOMIO-PROC)	
SN50.N	.KOMALT	. ALTERNATE PRINT FILE COUNTERS (SEE ALTPRT-PROCS)	
SN50.N	.KOMDET	. DETECTION FILE WINDOW PACKETS (SEE NERDET-PROCS)	
SN50.N	.KOMFIT	. ADJUSTMENT/REGISTRATION PARAMS (SEE NERDET-PROCS)	
SPRT.SC	DAH.KOMIO-PROC	. FORTRAN MANIPULATION OF ASSEMBLER I/O PACKETS	
SPRT.SC	DAH.KOMIRT-PROC	. COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS	
SN50.N	.KOMIHW	. COMMON INPUT WINDOW PACKETS (SEE WINDOW-PROCS)	
SN50.N	.KOMKLS	. COMMON CLASSIFICATION INFO (SEE NERDET-PROCS)	
SPRT.SC	DAH.KOMKS-PROC	. COMMON COLOR SCREEN PARAMETERS	
SN50.N	.KOML00	. L00 FILE I/O PKTS. POINTERS (SEE XQTL00-PROCS)	
SPRT.SC	DAH.KOMLUS-PROC	. COMMON I/O PACKET/POINTERS FOR UNIT 3	
SPRT.SC	DAH.KOMLUS-PROC	. COMMON BUFFER, POINTERS, FLAGS FOR UNIT 5	
SPRT.SC	DAH.KOML2N-PROC	. COMMON I/O PKTS FOR DETECTION FILES (UNITS 21-24)	
SN50.N	.KOMNER	. COMMON ERG SCENE PARAMETERS (SEE NERDET-PROCS)	
SPRT.SC	DAH.KOMNET-PROC	. COMMON CONTRL NETWORK COORDINATES	
SN50.N	.KOMOWW	. COMMON OUTPUT WINDOW PACKETS (SEE WINDOW-PROCS)	
SPRT.SC	DAH.KOMSLM-PROC	. COMMON SPECTRAL LIMITS	
SPRT.SC	DAH.KOMSYH-PROC	. COMMON SYMBOL TABLE	
SPRT.SC	DAH.KOMTBL-PROC	. COMMON MULTI-PURPOSE TABLE	
SN50.N	.KOMXQT	. COMMON PROGRAM EXECUTION INFO (SEE XQTL00-PROCS)	

DAM PACKAGE APPENDIX Q
MACROS

APPENDIX-Q
002

SPRT.SC DAM.KOMXQT-APROC	. COMMON PROGRAM EXECUTION INFO (ASH ROUINES ONLY
SPRT.SC DAM.LSTLUB-PROC	. NAMELIST SPECS FOR UNIT 0 (REGISTRATN PARAMETERS
SPRT.SC DAM.MAXBYT-PROC	. DEFINE MAXIMUM BYTE VALUE
SPRT.SC DAM.MAXICE-PROC	. DEFINE MAXIMUM INTEGER-CHAR-EQUIV VALUE
SPRT.SC DAM.MAXINT-PROC	. DEFINE MAXIMUM INTEGER VALUE
SPRT.SC DAM.NERDET-PROCS	. COMMON/HEADER BLOCKS FOR ERTS DETECTION FILES
BMSO.N .NITAB	. (SEE ALTPRT-PROCS)
BMSO.N .NITROT	. (SEE ATPRT-PROCS)
SPRT.SC DAM.NULCHR-PROC	. DEFINE NULL CHARACTER
SPRT.SC DAM.NULCHR-PROC	. DEFINE NULL CHARACTER STRING
SPRT.SC DAM.PICDEF-PROC	. DEFINE PICTAB PARAMETERS
BMSO.N .PPDL4C	. PPD LINE FOR CORRECTED (SEE TRFORM-PROCS)
BMSO.N .PPDC4C	. PPD COLUMN FOR CORRECTED (SEE TRFORM-PROCS)
SPRT.SC DAM.PRCDEF-PROC	. DEFINE PRTCLASS PARAMETERS
SPRT.SC DAM.PRCDEF-PROC	. DEFINE PRTDET PARAMETERS
SPRT.SC DAM.PXBDEF-PROC	. DEFINE STRUCTURE OF PIXEL BUFFER
BMSO.N .STM4C	. STM NORTHING FOR CORRECTED (SEE TRFORM-PROCS)
BMSO.N .SME4C	. STM EASTING FOR CORRECTED (SEE TRFORM-PROCS)
BMSO.N .SYSXQT	. (SEE XQTLOG-PROCS)
SPRT.SC DAM.TRFORM-PROCS	. DEFINE COORDINATE TRANSFORMATIONS
BMSO.N .WINDF	. (SEE WINDOW-PROCS)
SPRT.SC DAM.WINDF-PROCS	. COMMON/DEFINE FOR WINDOW PACKETS
BMSO.N .XQTDEF	. (SEE XQTLOG-PROCS)
SPRT.SC DAM.XQTLOG-PROCS	. COMMON BLOCKS FOR XQT & DAM LOG MANIPULATION

8PRT.SC DAM.PREFACE-R	. (8009)	SET TABS a 12 & 31
8PRT.SC DAM.APPENDIX-R	.	
8PRT.SC DAM.AS4CST	. ASCII BYTE STRING FOR CHARACTER STRING	
8PRT.SC DAM.AS4EB	. ASCII BYTE STRING FOR EBCDIC BYTE STRING	
8PRT.SC DAM.BST488/1108	. INTERNAL BYTE STRING FOR 8-BIT EXTERNAL BYTE STR	
8PRT.SC DAM.BST488/1110	. INTERNAL BYTE STRING FOR 8-BIT EXTERNAL BYTE STR	
8PRT.SC DAM.B848ST/1108	. 8-BIT EXTERNAL BYTE STRING FOR INTERNAL BYTE STR	
8PRT.SC DAM.B848ST/1110	. 8-BIT EXTERNAL BYTE STRING FOR INTERNAL BYTE STR	
8PRT.SC DAM.CBINIT	. INITIALIZE CHARACTER BUFFER	
8PRT.SC DAM.CB4CST	. CHARACTER BUFFER FOR CHARACTER STRING	
8PRT.SC DAM.CB4FIL	. CHARACTER BUFFER FOR FILE ('READ')	
8PRT.SC DAM.CB4IN	. CHARACTER BUFFER FOR INTEGER	
8PRT.SC DAM.CB4RL	. CHARACTER BUFFER FOR REAL	
8PRT.SC DAM.CLRQWD	. CLEAR QUARTER-WORD MODE	
8PRT.SC DAM.CST4AS	. CHARACTER STRING FOR ASCII BYTE STRING	
8PRT.SC DAM.CST4EB	. CHARACTER STRING FOR EBCDIC BYTE STRING	
8PRT.SC DAM.CST4IN	. CHARACTER STRING FOR INTEGER	
8PRT.SC DAM.CST4RL	. CHARACTER STRING FOR REAL	
8PRT.SC DAM.CURBST	. CURVATURE (1ST DERIVATIVE) OF BYTE STRING	
8PRT.SC DAM.CBS4CS	. VARIABLE-LENGTH (<= 8 CHAR) STRING FOR CHAR STR	
8PRT.SC DAM.CBS4IN	. VARIABLE-LENGTH (<= 8 CHAR) STRING FOR INTEGER	
8PRT.SC DAM.CBS4RL	. VARIABLE-LENGTH (<= 8 CHAR) STRING FOR REAL	
8PRT.SC DAM.DCODE	. DECODE NUMERIC CHARACTER STRING	
8PRT.SC DAM.EB4AS	. EBCDIC BYTE STRING FOR ASCII BYTE STRING	
8PRT.SC DAM.EB4CST	. EBCDIC BYTE STRING FOR CHARACTER STRING	
8PRT.SC DAM.FIL4CB	. FILE FOR CHARACTER BUFFER ('WRITE')	
8PRT.SC DAM.GETBYT	. GET NON-NEG INTEGER FROM BYTE IN BYTE STRING	
8PRT.SC DAM.GETCHR	. GET CHARACTER FROM CHARACTER STRING	
8PRT. : DAM.GETOBY	. GET NON-NEG INTEGER FROM DOUBLE BYTE IN BYTE STR	
8PRT. : DAM.GETHEX	. GET HEXADECIMAL CHAR FROM NYBLE IN BYTE STRING	
G.N .GETICE	. GET INTEGER-CHAR-EQUIV FROM CHAR STR (SEE GETCHR)	
8PRT.SC DAM.GETINT	. GET INTEGER FROM INTEGER STRING	
8PRT.SC DAM.GETNUL	. BEGIN ERROR WALKBACK (ARGS MATCH GETBYT/CHR/INT)	
8PRT.SC DAM.GETNYB	. GET NON-NEG INTEGER FROM NYBLE IN BYTE STRING	
8PRT.SC DAM.GETQBY	. GET INTEGER FROM QUADRUPLE BYTE IN BYTE STRING	
8PRT.SC DAM.GETOKH	. GET NEXT CHAR STRING DATA FLD FROM IMAGE BUFFER	
8PRT.SC DAM.GRABSA	. GRADIENT (1ST DERIVATIVE) OF BYTE STRING ARRAY	
8PRT.SC DAM.GTBYTS	. GET ARRAY OF NON-NEG INTEGERS FROM BYTE STRING	
8PRT.SC DAM.ICE	. INTEGER CHARACTER EQUIVALENT (FROM CHARACTER)	
8PRT.SC DAM.ICHR	. CHARACTER (FROM INTEGER CHARACTER EQUIVALENT)	
8PRT.SC DAM.IVKONE	. INTEGER FOR ONE'S COMPLEMENT	
8PRT.SC DAM.IVKTWO	. INTEGER FOR TWO'S COMPLEMENT	
8PRT.SC DAM.KHR4IN	. * ENCODE 8-CHAR (FIELDATA) STRING FROM INTEGER	
8PRT.SC DAM.KONE4I	. ONE'S COMPLEMENT FOR INTEGER	
8PRT.SC DAM.KTWO4I	. TWO'S COMPLEMENT FOR INTEGER	
8PRT.SC DAM.LAPBS-	. LAPLACIAN (2ND DERIVATIVE) OF BYTE STRING ARRAY	
8PRT.SC DAM.LBYTEQ	. LOCATION OF BYTE IN STRING EQUAL TO SEARCH BYTE	
8PRT.SC DAM.LBYTNE	. LOCATION OF BYTE IN STRING NOT EQ TO SEARCH BYTE	
8PRT.SC DAM.LCHREQ	. LOCATION OF CHAR IN STRING EQUAL TO SEARCH CHAR	
8PRT.SC DAM.LCHRNE	. LOCATION OF CHR IN STRING NOT EQ TO SEARCH CHR	
8PRT.SC DAM.LCSTEQ	. LOCATION IN ONE CHARACTER STRING OF ANOTHER	
8PRT.SC DAM.LENCST	. LENGTH OF CHARACTER STRING	
8PRT.SC DAM.LENPAD	. LENGTH PADDED TO NEXT WORD BOUNDARY	
8PRT.SC DAM.LICEEQ	. LOCATION OF ICE IN STRING EQUAL TO SEARCH ICE	
8PRT.SC DAM.LICENE	. LOCATION OF ICE IN STRING NOT EQ TO SEARCH ICE	
8PRT.SC DAM.LINTEQ	. LOCATION OF INTEGER IN STRING EQ TO SEARCH INT	

DAM PACKAGE APPENDIX R
CHAR/BYTE/STRING ROUTINES

APPENDIX-R
002

SPRT.SC DAM.LINTNE	. LOCATION OF INTEGER IN STRING NE TO SEARCH INT
SPRT.SC DAM.LOWCST	. # OF CHAR STRING LOWER IN COLLATING SEQUENCE
SPRT.SC DAM.MOVBST/ASM	. MOVE BYTE STRING
SPRT.SC DAM.MOVBST/FOR	. MOVE BYTE STRING
SPRT.SC DAM.MOVBYT	. MOVE BYTE
SPRT.SC DAM.MOVCHR	. MOVE CHARACTER
SPRT.SC DAM.MOV CST/ASM	. MOVE CHARACTER STRING
SPRT.SC DAM.MOV CST/FOR	. MOVE CHARACTER STRING
SPRT.SC DAM.MOVDBY	. MOVE DOUBLE BYTE
SPRT.SC DAM.MOVIST	. MOVE INTEGER STRING
SPRT.SC DAM.NB4NI	. NUMBER OF BYTES FOR NUMBER OF INTEGERS
SPRT.SC DAM.NC4NI	. NUMBER OF CHARACTERS FOR NUMBER OF INTEGERS
SPRT.SC DAM.NEXTOK	. GET POINTERS TO NEXT TOKEN IN IMAGE BUFFER
SPRT.SC DAM.NI4NB	. NUMBER OF INTEGERS FOR NUMBER OF BYTES
SPRT.SC DAM.NI4NC	. NUMBER OF INTEGERS FOR NUMBER OF CHARACTERS
SPRT.SC DAM.PUTBYT	. PUT NON-NEG INTEGER INTO BYTE OF BYTE STRING
SPRT.SC DAM.PUTCHR	. PUT CHARACTER INTO CHARACTER STRING
SPRT.SC DAM.PUTDBY	. PUT NON-NEG INTEGER INTO DOUBLE BYTE OF BYTE STR
SPRT.SC DAM.PUTHEX	. PUT HEXADEXIMAL CHAR INTO NYBLE IN BYTE STRING
MSO.N .PUTICE	. PUT INTEGER-CHAR-EQUIV INTO CHAR STR (SEE PUTCHR
SPRT.SC DAM.PUTINT	. PUT INTEGER INTO INTEGER STRING
SPRT.SC DAM.PUTNYB	. PUT NON-NEG INTEGER INTO NYBLE OF BYTE STRING
SPRT.SC DAM.PUTQBY	. PUT INTEGER INTO QUADRUPLE BYTE OF BYTE STRING
SPRT.SC DAM.SETQWD	. SET QUARTER-WORD MODE
SPRT.SC DAM.SLOBST	. SLOPE (1ST DERIVATIVE) OF BYTE STRING
SPRT.SC DAM.TRUEAL	. TRUE IF CST IS ALPHA (26 LTRS + SPACE)
SPRT.SC DAM.TRUCST	. TRUTH VALUE OF CHARACTER STRING COMPARISON

DAM PACKAGE APPENDIX S
SORT ROUTINES

APPENDIX-S
001

SPRT.SC DAM.PREFACE-S	. (0009)	SET TABS 8 12 & 31
SPRT.SC DAM.APPENDIX-S	.	
SPRT.SC DAM.ISRTBA	. INTEGER BUBBLE SORT ASCENDING	
SPRT.SC DAM.ISRTBD	. INTEGER BUBBLE SORT DESCENDING	
SPRT.SC DAM.ISRTHA	. INTEGER HIBBARD'S SHELLSORT ASCENDING	
SPRT.SC DAM.ISRTHD	. INTEGER HIBBARD'S SHELLSORT DESCENDING	
SPRT.SC DAM.ISRTSA	. INTEGER SHUTTLE SORT ASCENDING	
SPRT.S M.ISRTSD	. INTEGER SHUTTLE SORT DESCENDING	
SPRT.S M.ISRTHS	. TAGSORT USING HIBBARD'S SHELLSORT	