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.
DETECTION AND MAPPING (DAM) PACKAGE

Volume 4a: Software System Manual (part 1)

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
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

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


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 multispectral 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", "A/F", 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

JSC Form 1424 (Rev Nov 75)

NASA — JSC
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.
(This volume contains Appendices I thru M)
<table>
<thead>
<tr>
<th>SPRT.SC</th>
<th>NAME</th>
<th>APPENDIX-A (0009)</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAN</td>
<td>DAN PACKAGE</td>
<td>A-3</td>
<td></td>
</tr>
<tr>
<td>EXP</td>
<td>DBM</td>
<td>A-4</td>
<td></td>
</tr>
<tr>
<td>DAN</td>
<td>DBM</td>
<td>A-5</td>
<td></td>
</tr>
<tr>
<td>EXP</td>
<td>DBM</td>
<td>A-6</td>
<td></td>
</tr>
<tr>
<td>RULES</td>
<td>RULES</td>
<td>A-9</td>
<td></td>
</tr>
<tr>
<td>FILES</td>
<td>FILES</td>
<td>A-10</td>
<td></td>
</tr>
<tr>
<td>CARD</td>
<td>CARD CODES</td>
<td>A-11</td>
<td></td>
</tr>
<tr>
<td>COORDINA</td>
<td>COORDINATES</td>
<td>A-12</td>
<td></td>
</tr>
<tr>
<td>COMPUTER</td>
<td>COMPUTER RUNS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCAL</td>
<td>LOCAL STANDARDS</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DAN PACKAGE APPENDIX B
EXEC COMMAND DOCUMENTATION

APPENDIX-B (0008) B-1

0PRT.SC DAM.
0PRT.SC DAM.EXP----
0PRT.SC DAM.EXP-EXEC
0PRT.SC DAM.EXP-SADD
0PRT.SC DAM.EXP-SASS
0PRT.SC DAM.EXP-SCOPY
0PRT.SC DAM.EXP-SDATA/CHECKOUT
0PRT.SC DAM.EXP-SED
0PRT.SC DAM.EXP-SEND
0PRT.SC DAM.EXP-SEOF
0PRT.SC DAM.EXP-SFIN
0PRT.SC DAM.EXP-SFREE
0PRT.SC DAM.EXP----
0PRT.SC DAM.EXP-SLOCATE
0PRT.SC DAM.EXP----
0PRT.SC DAM.EXP----
0PRT.SC DAM.EXP----
0PRT.SC DAM.EXP-SFREE
0PRT.SC DAM.EXP-SFREE
0PRT.SC DAM.EXP-SFREE
0PRT.SC DAM.EXP-SSWAP
0PRT.SC DAM.EXP-SXQT
0PRT.SC DAM.EXP----
<table>
<thead>
<tr>
<th>SPRT.SC</th>
<th>DAN.EXP</th>
<th>APPENDIX-C</th>
<th>( 0099 )</th>
<th>C-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-PROGRAMS</td>
<td>SUMMARY</td>
<td>C-3</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-ERTS-DUP</td>
<td>ERTS-DUP</td>
<td>C-4</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-ERTSIDC</td>
<td>ERTSIDC</td>
<td>C-5</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-PICTAB</td>
<td>PICTAB</td>
<td>C-6</td>
<td></td>
</tr>
<tr>
<td>MORE:</td>
<td></td>
<td></td>
<td>C-7</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-CONTROL</td>
<td>CONTROL</td>
<td>C-8</td>
<td></td>
</tr>
<tr>
<td>MORE:</td>
<td></td>
<td></td>
<td>C-9</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-CLASSIFY</td>
<td>CLASSIFY</td>
<td>C-10</td>
<td></td>
</tr>
<tr>
<td>MORE:</td>
<td></td>
<td></td>
<td>C-11</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-PRDDET</td>
<td>PRTDET</td>
<td>C-12</td>
<td></td>
</tr>
<tr>
<td>MORE:</td>
<td></td>
<td></td>
<td>C-13</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-PRTCCLASS</td>
<td>PRTCLASS</td>
<td>C-14</td>
<td></td>
</tr>
<tr>
<td>MORE:</td>
<td></td>
<td></td>
<td>C-15</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP-PLTCCLASS</td>
<td>PLTCCLASS</td>
<td>C-16</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC</td>
<td>DAN.EXP</td>
<td>STATUS</td>
<td>C-17</td>
<td></td>
</tr>
</tbody>
</table>

C-1
<table>
<thead>
<tr>
<th>Command</th>
<th>APPENDIX-0</th>
<th>(GOOD)...D-1.</th>
<th>SET TABS...E7 &amp; 98</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPRT.SC DAN.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MORE:</td>
<td>0-2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-COMANDS</td>
<td>SUMMARY</td>
<td>0-3</td>
<td></td>
</tr>
<tr>
<td>MORE:</td>
<td>0-5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MORE:</td>
<td>0-6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td></td>
<td>(FUTURE)</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-ADJ</td>
<td>ADJUST</td>
<td>0-7</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-ALI</td>
<td>ALIGN</td>
<td>0-8</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>ATTITUDE</td>
<td>0-9</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-CHN</td>
<td>CHANNEL</td>
<td>0-10</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-CEN</td>
<td>CENTER</td>
<td>0-11</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-COL</td>
<td>COLOR</td>
<td>0-12</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-COP</td>
<td>COPIES</td>
<td>0-13</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-COU</td>
<td>COUNT</td>
<td>0-14</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-CSC</td>
<td>CROSSTAB</td>
<td>0-15</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>DENSITY</td>
<td>0-16</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-DET</td>
<td>DETECT</td>
<td>0-17</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-DIA</td>
<td>DIAGRAM</td>
<td>0-18</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-DIS</td>
<td>DISPLAY</td>
<td>0-19</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-EX</td>
<td>EXIT</td>
<td>0-20</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-EXP</td>
<td>EXPLAIN</td>
<td>0-21</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-FAC</td>
<td>FACTOR</td>
<td>0-22</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-FL</td>
<td>F. (ENDIF)</td>
<td>0-23</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-GE</td>
<td>GEOMETRY</td>
<td>0-24</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>HISTOGRAM</td>
<td>0-25</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>HISTOGRAM</td>
<td>0-26</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-IF</td>
<td>IF</td>
<td>0-27</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-INT</td>
<td>INTENSITY</td>
<td>0-28</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-LIN</td>
<td>LINEAR</td>
<td>0-29</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-LIS</td>
<td>LIST</td>
<td>0-30</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-MAP</td>
<td>MAP</td>
<td>0-31</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-MER</td>
<td>MERIDIAN</td>
<td>0-32</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>MODE</td>
<td>0-33</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-NAM</td>
<td>NAME</td>
<td>0-34</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-NEW</td>
<td>NEWS</td>
<td>0-35</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-LEX</td>
<td>NEXT</td>
<td>0-36</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-OF</td>
<td>OFF</td>
<td>0-37</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-ON</td>
<td>ON</td>
<td>0-38</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-ORI</td>
<td>ORIGIN</td>
<td>0-39</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>PAGE</td>
<td>0-40</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-FAD</td>
<td>PAG4</td>
<td>0-41</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP---</td>
<td>PAG4</td>
<td>0-42</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-PIC</td>
<td>PICTURE</td>
<td>0-43</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>PLOTTER</td>
<td>0-44</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-POI</td>
<td>POINT</td>
<td>0-45</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-POK</td>
<td>POKE</td>
<td>0-46</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-POK</td>
<td>POKE</td>
<td>0-47</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-POK</td>
<td>POKE</td>
<td>0-48</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-PRO</td>
<td>PRINTER</td>
<td>0-49</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP-PRO</td>
<td>PROFILE</td>
<td>0-50</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>PRINT</td>
<td>0-51</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>PRINT</td>
<td>0-52</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>PRINT</td>
<td>0-53</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>PRINT</td>
<td>0-54</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>PRINT</td>
<td>0-55</td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAN.EXP----</td>
<td>PRINT</td>
<td>0-56</td>
<td></td>
</tr>
</tbody>
</table>

APPENDIX-0  081
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPRT.SC OAN.EXP-RAN</td>
<td>RANK</td>
<td>0-87</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-REN</td>
<td>RENUMBER</td>
<td>0-89</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-RES</td>
<td>RESAMPLING</td>
<td>0-90</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-ROT</td>
<td>ROTATE</td>
<td>0-90</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-SCA</td>
<td>SCALE</td>
<td>0-91</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-SCC</td>
<td>SCENCE</td>
<td>0-92</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-SHA</td>
<td>SHARPENING</td>
<td>0-94</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-SIZ</td>
<td>SIZE</td>
<td>0-98</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-SPA</td>
<td>SPACING</td>
<td>0-98</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-SYN</td>
<td>SPHEROID</td>
<td>0-99</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-SYM</td>
<td>SYMBOLS</td>
<td>0-70</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-TAB</td>
<td>TABULATE</td>
<td>0-71</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-TIC</td>
<td>TICK</td>
<td>0-73</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-TIN</td>
<td>TIME</td>
<td>0-74</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-TOL</td>
<td>TINT</td>
<td>0-75</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-TOL</td>
<td>TOLERANCE</td>
<td>0-78</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-TOT</td>
<td>TOTAL</td>
<td>0-77</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-MIN</td>
<td>WINDOW</td>
<td>0-79</td>
</tr>
<tr>
<td>SPRT.SC OAN.EXP-ZON</td>
<td>ZONE</td>
<td>0-80</td>
</tr>
<tr>
<td>SPRTR. SC</td>
<td>DAN. APPENDIX-F</td>
<td>(0000)</td>
</tr>
<tr>
<td>-----------</td>
<td>----------------</td>
<td>--------</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1037-16244</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1037-16244/UTH</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1073-16244-3</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1062-16309</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1062-16309-3</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1132-16812</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1291-16381</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1299-16499</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1299-16618</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1299-16201</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1302-16899</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1407-15381</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1426-18209</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1426-18309</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1704-18231</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 1760-20381</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 21060-17613</td>
</tr>
<tr>
<td></td>
<td>SPRTR. SC</td>
<td>DAN. 30130-18032</td>
</tr>
</tbody>
</table>
CAN PACKAGE APPENDIX 0
SPECTRAL LIMITS/TRANSFORMS

SPRT.SC CAN.TURBID-NESS/1 .0-00
SPRT.SC CAN.TURBID-NESS/2A .0-00
SPRT.SC CAN.TURBID-NESS/2B .0-00
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPRT.SC DAN.</td>
<td>Sample Runstreams</td>
</tr>
<tr>
<td>OPRT.SC DAN.EXP----</td>
<td></td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-ERTS-DUP</td>
<td></td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-ERTS-DUP/JSC</td>
<td></td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-PICTAB/X-SF</td>
<td>PICTAB(SINGLE FILE 'X' TAPES)....H-7</td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-PICTAB/X-MF</td>
<td>PICTAB(MULTI-FILE 'X' TAPES)....H-8</td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-PICTAB/PH</td>
<td>PICTAB('PH' TAPES)..............H-9</td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-CONTROL</td>
<td>CONTROL............................H-11</td>
</tr>
<tr>
<td>OPRT.SC DAN.EXP----</td>
<td></td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-CLA-PRC/X</td>
<td>CLASSIFY/PRCCLASS('X' TAPES)....H-13</td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-CLA-PRC/PH</td>
<td>CLASSIFY/PRCCLASS('PH' TAPES)....H-14</td>
</tr>
<tr>
<td>OPRT.SC DAN.EXP----</td>
<td></td>
</tr>
<tr>
<td>OPRT.SC DAN.RUN-STATUS</td>
<td>STATUS..............................H-16</td>
</tr>
</tbody>
</table>
DAN PACKAGE APPENDIX 1
REVISIONS AND NEWS

SPRT.SC OAH.APPENDIX-1
SPRT.SC OAH.REV-OAH
SPRT.SC OAH.NEW-OAH
SPRT.SC OAH.NEW-PLTCCLASS
DAN PACKAGE APPENDIX I

REVISIONS AND NEWS

07808: PLTCLASS NOT AVAILABLE (USE VERSION 8-08)
07808: ACCEPT JULY '77 TAPE ANNOTATION RECORD
07808: EXTRACT ATTITUDE & CALIBRATION DATE FROM SIAT FILE
07808: ACCEPT JAN '78 TAPE 10 RECORD
07802: LANDSAT-2 WATER LIMITS & TRANSFORMS
07804: REVISED WATER LIMITS & TRANSFORMS
07804: FIXES FOR UNIVAC PRINTER I/O BUGS
07807: REVISED PRINTER & DENSITY COMMANDS
07807: NEW OITCOP PROGRAM
07807: NEW TOLERANCE COMMAND
07807: MORE OFF/ON MODE OPTIONS
07807: LARGER DISPLAY/MAP PRINT FILES
07807: FIX DETECTION FILE READING BUFFER LINE ERROR
07803: STATUS FINDS BACKLOG & ACTIVE RUNS
07803: CLASSIFY MARKS DETECTION FILE BAD AFTER FATAL TAPE ERROR
07803: MACRO COMMANDS
07803: DEFAULT/MACRO COMMANDS MAY BE IN PRIVATE USER FILE
07803: COUNT & TIME COMMANDS
07803: DETECT.RADIANCE
07803: REV WATER LIMITS & TRANSFORMS FOR VERY HIGH SUN ANGLE
07912: PX90EF INTERNAL BUFFER STRUCTURE
07912: HANDLE LANDSAT-'2/-3 BIP 'AM' & 'PM' TAPES
07912: HANDLE LANDSAT-3 BIP 'X' TAPES
07912: NEGATIVE RADIANCE IN TRANSFORMED CHANNEL CONVERTED TO ZERO
07912: RANGE FOR RADIANCE DETECTION FILES EXPANDED TO 0-127
07912: NON-INTEGER SPACING
07912: GEOMETRY, CENTER, SIZE, LIST COMMANDS
07912: IF, FI, PEEK, POKE, COLOR, INTENSITY, PICTURE COMMANDS
07912: LANDSAT-20 ('PM' TAPES) WATER LIMITS & TRANSFORMS
08005: MULTI-CHANNEL RADIANCE DETECTION FILES
08005: PRDENS CHANGED TO PRDST
08005: OPTIONAL DISK INPUT TO PICTAB & CLASSIFY
08005: COLOR GRAPHICS INTERFACE
08005: LANDSAT-3A & -3C WATER LIMITS & TRANSFORMS
08009: TAPE QWD STATUS FROM FITEMS OR INFOS
08009: FIX MSKPIX BIN TYPE ERROR WHEN OVER 2 VERTICES

0

07808: IN BATCH, INSERT SPR.T BETWEEN BUSE & BASO. AS BELOW:
   BUSE DAN. (NAME OF DAN FILE)
   SPR.T
   BASO A DAN.

ORIGINAL PAGE IS OF POOR QUALITY
DAN PACKAGE APPENDIX I
REVlS10N S AND NEWS

1.

PLTCLASS NEWS

VERSION 7002:
I. PLTCLASS NOT CURRENTLY SUPPORTED (USE VERSION 5-06)
DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

SPRT.SC DAN.Appendix-J
SPRT.SC DAN.DEF-ERTSIDC
SPRT.SC DAN.DEF-PICTAB
SPRT.SC DAN.DEF-CONTROL
SPRT.SC DAN.DEF-CLASSIFY
SPRT.SC DAN.DEF-PROTECT
SPRT.SC DAN.PRO-DEF-RAD
SPRT.SC DAN.PRO-DEF-DEF
SPRT.SC DAN.PRO-DEF-CLA
SPRT.SC DAN.DEF-PRTCCLASS
SPRT.SC DAN.DEF-PRO-DEF-
SPRT.SC DAN.DEF-DEF-DEF
SPRT.SC DAN.DEF-DEF-CLA
SPRT.SC DAN.DEF-PITCLASS
SPRT.SC DAN.DEF-STATUS
DAN PACKAGE APPENDIX J
DEFAULT COMMANDS

*there are no default commands for ertsoc
OAN PACKAGE APPENDIX J
DEFAULT COMMANDS

ON.DEFAULT..PICTAB DEFAULT COMMANDS
NAME. ... (NO NAME)
LINEAR.1.HEIGHTS.1.0.0.0 ... LINEAR.1.GAIN.1.BIAS.-9 ... (CHAN 1) - 0
LINEAR.2.HEIGHTS.0.0.0.1 ... LINEAR.2.GAIN.1.BIAS.0 ... (CHAN 4)
POLAR.1.GAIN.1.BIAS.0 ... POLAR.2.GAIN.57.290 DEG/RADIAN.BIAS.0
PRINTER.FILES.1
ON.PROMPT.CONFIRM.LEGEND
COPIES.1
SYN.0.0.0.0...SYN.A.10.J.10...SYN.K.20.T.30...SYN.U.40.Z.187
CHANNEL.4
RADIANCE.0.19
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.-6.-25.+8.+25..1:LIN X 51COL...SPA.3.2...FI
IF.ON.BATCH...MIN.PRINT.-180.-60.+180.+60..361X121...SPA.1.1...ON.ECHO...FI
OFF.DEFAULT..PICTAB
ON. DEFAULT..CONTROL DEFAULT COMMANDS
ON. PROMPT. CONFIRM
OFF. DEFAULT..CONTROL
ON.DEFAULT..CLASSIFY DEFAULT COMMANDS
TOLERANCE.0
ON.PROMPT.CONFIRM
COPIES:1
IF.OFF.BATCH..MIN.SCAN.-00.-400.-80..400..1801N X 0015AN...F1
IF.ON.BATCH..ORIG.SCAN.1..MAX.SCAN.2500..3500..ENTIRE SCENE...ON.ECHO...F1
OFF.DEFAULT..CLASSIFY
OFF.DEFAULT ... *** PRTOET DEFAULT COMMANDS ***
PRINTER.FILES.1
ON.PROMPT.CONFIRM.LEGEND
COPIES.1
MACSPEC-1-2.DEF.(DETECT) .. DEFAULTS FOR CURRENT TYPE OF DETECTION FILE
SPACING.1.1
ZONE
IF.OFF.BATCH ... WINDOW.PRINT. -8.-29. -8. .29. 17 LINE X 61 COL ... F1
IF.ON.BATCH ... WINDOW.PRINT. -100.-60. +100.+60. 301X181 ... ON.ECHO ... F1
OFF.DEFAULT..PRTOET
DEF-RAD ... PDTDET MACRO COMMAND WITH DEFAULTS FOR RADIANCE DETECTION FILE
RADIANCE.0.127
STR.0.0.0.0...STR.A.18.J.19...STR.K.20.T.30...STR.U.40.Z.127
### DEFAULT COMMANDS

**DEF-DEN** . . . **PRDSET MACRO COMMAND WITH DEFAULTS FOR DENSITY DETECTION FILE**

**DENSCITY.G.19**

**SYN.G.G.S.G..SYN.A.10.J.10**

<table>
<thead>
<tr>
<th>NUMBER OF</th>
<th>CENTRAL PIXEL CLASSIFICATION STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEIGHBORING</td>
<td></td>
</tr>
<tr>
<td>PIXELS OF</td>
<td>OTHER CLASS</td>
</tr>
<tr>
<td>NAMED CLASS</td>
<td>------------</td>
</tr>
<tr>
<td>------------</td>
<td>------------</td>
</tr>
<tr>
<td></td>
<td>0 00 0 16 A 20</td>
</tr>
<tr>
<td>1 01 1 11 B 21</td>
<td></td>
</tr>
<tr>
<td>2 03 3 13 D 23</td>
<td></td>
</tr>
<tr>
<td>3 05 5 15 F 25</td>
<td></td>
</tr>
<tr>
<td>4 07 7 17 H 27</td>
<td></td>
</tr>
<tr>
<td>5 09 9 19 J 29</td>
<td></td>
</tr>
</tbody>
</table>
DEF-CLA  .. PRTDET MACRO COMMAND WITH DEFAULTS FOR CLASS DETECTION FILE
CLASS.G.61
ON. DEFAULT..PRCLASS DEFAULT COMMANDS
ON. PROMPT. CONFIRM. LEGEND
COPIES 1
MACSPEC-1-2.DEF.(DETECT) .. DEFAULTS FOR CURRENT TYPE OF DETECTION FILE
SCALE 1/24000
WINDOW MINUTES 7.9.7.5
TICK INTERVAL MINUTES 7.5.7.5 MINUTES 2.5.2.5
ZONE
IF. ON. BATCH ... ON. ECHO ... OFF. CONFIRM ... FI
OFF. DEFAULT..PRCLASS
DEF-RAD...PRICLASS MACRO COMMAND WITH DEFAULTS FOR RADIANCE DETECTION FILE
RADIANCE.0.127
COUNT.0
SYN.0.0.9...SYN.A.10.J.19...SYN.K.20.T.39...SYN.U.40.Z.127
DEF-GEN  ... PRICLASS MACRO COMMAND WITH DEFAULTS FOR DENSITY DETECTION FILE
DENSITY.10.19
COUNT.1
SYN. 0...SYN.1...SYN.9...SYN.A.10...SYN.R.99...SYN.S.100...SYN.Y.109...SYN.Z.170.255
DEF-CLA  ... PRTCLASS MACRO COMMAND WITH DEFAULTS FOR CLASS DETECTION FILE
CLASS.0.81
COUNT.0
SYM.0.0.2.8...SYM.A.10.J.19...SYM.K.20.T.39...SYM.U.40.Z.81
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
ON.DEFAULT..STATUS
ON.PROMPT.CONIRM
NEXT.ON.BATCH ... ON.ECHO
OFF.DEFAULT..STATUS
THE OAM PACKAGE SOFTWARE IS DESIGNED TO RUN ON UNIVAC 1100 COMPUTERS
UNDER THE EXEC-B 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-B COMMAND LANGUAGE, AND EXEC-B 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 'P' 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 G) 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).
APPENDIX-K
SYSTEM IMPLEMENTATION

SPRT.SC OAM.PREFACE-K
SPRT.SC OAM.APPENDIX-K
SPRT.SC OAM.SYS-HIA
MSGO.N OAM.SYS-TITLE
SPRT.SC OAM.SYS-LIST
SPRT.SC OAM.SYS-EXPLAIN
SPRT.SC OAM.SYS-COPYCOM
SPRT.SC OAM.SYS-FOROPT
SPRT.SC OAM.SYS-BLOCK
SPRT.SC OAM.SYS-NAPOPT
SPRT.SC OAM.SYS-COMPILE
SPRT.SC OAM.SYS-COLLECT
SPRT.SC OAM.SYS-DELETE
SPRT.SC OAM.SYS-GENCOM

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

ORIGINAL PAGE 1B
OF POOR QUALITY
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-B 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:

```plaintext
USE DAM..<QUALIFIER><FILE>
(SPRT.T1) (FIX FOR EXEC SYNC BUG)
BASE.DAM.
ADD DAM.SYS-LIST
```

2.2 THE DAM PACKAGE IS WRITTEN IN UNIVAC EXEC-B CONTROL LANGUAGE, EXTENDED UNIVAC FORTRAN IV (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:

```plaintext
USE DAM..<QUALIFIER><FILE>
(SPRT.T1) (FIX FOR EXEC SYNC BUG)
BASE.DAM.
(LOCAL CHANGES TO DAM.SYS-COLLECT) (SEE 2.4)
(LOCAL CHANGES TO MACROS) (SEE 5.2)
(LOCAL CHANGES TO DAM.SYS-FOREOPT) (SEE 4.1)
ADD DAM.SYS-COPYCOM
ADD COMPIL (ADD DELETE) (SEE 2.3)
(LOCAL CHANGES TO DAM.SYS-BLOCK) (SEE 4.0)
(LOCAL CHANGES TO DAM.SYS-MAPOPT) (SEE 4.1)
ADD COLLECT
(LOCAL CHANGES TO DAM.NEW-DAM) (SEE 4.8)
(LOCAL CHANGES TO DEFAULT COMMANDS) (SEE 5.0)
```

2.3 THE DAM.SYS-COPYCOM ELEMENT COPIES COMMAND STREAMS FROM THE DAM FILE TO
THE FOLLOWING 3 ELEMENTS IN TPFS (THEY MUST NOT BE SADD-ED DIRECTLY FROM THE DAM FILE SINCE THEY SACK IT):

TPFS.COMPILE - CONTAINS SPDP. SDELETE.A. SFOR. SASH STATEMENTS. AS APPROPRIATE FOR ALL PROCES. PROGRAMS. SUBROUTINES.
TPFS.DELCITE - CONTAINS SDELETE.B STATEMENTS FOR ALL SOURCE PROGRAMS AND SUBROUTINES (OTHER SYMBLO 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.
TPFS.Collect - Copied FROM DAM.SYS-COLLECT (SEE BELOW)

2.4 THE DAM.SYS-COLLECT ELEMENT CONTAINS ONE MAP STATEMENT FOR EVERY PROGRAM IN THE DAM PACKAGE. ANY MAP STATEMENT SPECIFYING THE 'VIRTUAL' VERSION OF THE MAP SOURCE ELEMENT COLLECTS A VIRTUAL OR 'SKELETON' ABSOLUTE ELEMENT WHICH. WHEN SADD-ED. TRIGGERS THE COLLECTION IN TPFS 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: 6RUN.<PRIORITY>/HR <RUNID. ETC.>
SUBE DAM..<QUALIF1ER><<FILE>
ISPRT.T1
8ASS.A DAM.
SADD DAM.SETUP
PROGRAMS SHOULD THEN BE EXECUTED FROM TPFS. 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 SADD-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 8. IF PARITY ERRORS OCCUR IN READING ORIGINAL CETS TAPES. COPIES SHOULD BE MADE WITH DAM.ENTS-DUP. AND THE COPIES USED. PROGRAMS IN A REMOTE BATCH RUN "AY 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-HAPOPT 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 NATHPACK ROUTINES ARE NOT IN THE SYSTEM RELOCATABLE LIBRARY. THEN CODE SIMILAR TO THE FOLLOWING MUST BE ADDED TO DAM.SYS-HAPOPT:

LIB DAM.
LIB <NAME OF FILE WITH FORTRAN V (FIELDATA) LIBRARY RELOCATABLES>
LIB <NAME OF FILE WITH NATHPACK RELOCATABLES>
LIB DAM.
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 $HALT$ in DAM.SYS-BLOCK must always equal the installation exec-$O$ system generation parameter $FU$ (maximum allowable number of active alternate print files). The value assigned to $HALT$ in DAM.SYS-BLOCK represents the default number of alternate print files to be used by programs in the DAM package. $HALT$ should generally correspond to the width in line-printer pages of the widest map that normally will be generated (4 is typical). However, $HALT$ must never exceed 9 and it must never exceed 91. (The 'PRINTER' command can be used within a program to temporarily change $HALT$ but not $FU$).

4.3 The device-type-mnemonic used to designate on-site line printer(s) should be assigned to $MATH$ 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 $LSPAGE$ (system print lines per page) in DAM.SYS-BLOCK must exactly match the installation standards.

4.5 It is highly desirable (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. $LINCH$ (DAM print lines per inch) and $LSPAGE$ (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 $MATH$ (see 4.3) then the value assigned to $KINCH$ in DAM.SYS-BLOCK must be 'NOM', otherwise the value must be 'NOM' 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 $KINCH$ (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 $MATH$ (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)$<\text{FILE}>$ and FILE size in sectors of the LOG file, specified by LOGFIL and LOGSEC, must be assigned in DAM.SYS-BLOCK. If LOGSEC 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.NEM-DAM should be revised as appropriate, subject to the following: This new 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:

9.1 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 SGN AT A LOCAL BOX PAGE UTILITY PROGRAM AT THE BEGINNING OF EACH BATCH RUN.

9.2 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 KTABLE PARAMETER STATEMENT IN THE KONTBL PROC (ELEMENT FOR PROCS).

KSYM - THIS ARRAY STORES CHARACTER SYMBOLS FOR EACH MAP. ONE SYMBOL GROUP PER WORD. IF MAP SCALES MUCH SMALLER THAN 1/256,000 ARE TO BE USED THIS ARRAY MUST BE ENLARGED. TO DO THIS CHANGE THE KSYM SIZE PARAMETER STATEMENT IN KONSYM-PROC (APPENDIX-O).

9.3 MOST PROGRAMS IN THE DAN PACKAGE USE DEFAULT COMMANDS CONTAINED IN SYmbOLIC ELEMENTS NAMED AS FOLLOWS:

DAN.DEF<(PROGRAM NAME) DEFAULT COMMANDS MAY BE CHANGED WITHOUT RECOMPIILING.

6.0 CONFLICTS BETWEEN RUNS

SEVERAL PROGRAMS IN THE DAN PACKAGE DYNAMICALLY ASSIGN CATALOGED PSAFRAND 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:

*OAMPRX-0. THROUGH *OAMPRX-9.

*OANDET-1. THROUGH *OANDET-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 SYM-ED INTERNALLY. THE SYSTEMS ANALYST MAY THEN EXAMINE THESE FILES FROM A REMOTE TERMINAL WITH THE BEN PROCESSOR AND MUST MANUALLY SYM OR DELETE 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
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 EXECUTED:

```
SAS0.(OPTIONS) 10..U9.(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 31.244.2118
- FURPUR 0026
- PDPT10 RL70-6
- FOR 90E3
- ASH130 RL1889
- HAP27.1 RL71-3
- SYSRRLIBS. 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:

```
\%XQ STATUS
KEY.<WRITE KEY FOR THE DAM PROGRAM FILE>
///.9999
EXIT
```
A00 ELEMENT MUST NOT PACK ITS OWN FILE
A00 ELEMENT MUST NOT PACK ITS OWN FILE
A00 ELEMENT MUST NEVER PACK ITS OWN FILE
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

COMPILED (DATA=SHORT), (RETN=NOP), (1110=OPT)

SEOF

K-10
**DAM PACKAGE APPENDIX K**

**SYSTEM IMPLEMENTATION**

**BLOCK DATA**

```plaintext
C

INCLUDE SYXQT.LIST
DATA JPRCN'/H.X.1.'/  a PRCNS IMAGE - DO NOT CHANGE!
DATA JRUN'/BLOG': '/  a CSFS LOG - DO NOT CHANGE!
DATA JPROG/4*/'  a BLANK FILL - DO NOT CHANGE!
DATA (JHDS(N.1).N=1.12)  a HEADING LINE 1
L /12*/' /  a HEADING LINE 2
DATA (JHDS(N.2).N=1.12)
DATA LUNCH/81  a DAM LINES/INCH
DATA KINCH/10/  a DAM COLUMNS/INCH
DATA LPAGE/68/  a DAM LINES/PAGE
DATA KPAGE/132/  a DAM COLUMNS/PAGE
DATA LSINCH/8/  a INSTALLATION SYS-GEN LINES/INCH
DATA LSPAGE/68/  a INSTALLATION SYS-GEN LINES/PAGE
DATA KNPRT/'NON'/  a PRINT CONTROL: 'AUT'/'MAN'/'NON'
DATA MNEMON/'PR'/  a DEVICE TYPE MNEMONIC FOR ONSITE PRINTERS
DATA NALTH/4/  a DAM NUMBER OF ALT PRT FILES (HALTH <= MSALTH)
DATA MSALTH/4/  a SYS GEN PARAM MSALTH (MAX NO ALT PRT FILES)

C

INCLUDE KOMLOG.LIST
DATA LQPKT/'1.','.4+0/'  a PACKET - DO NOT CHANGE!
DATA LQMSC/900/  a NUMBER OF SECTORS IN LOG FILE --
DATA LQSF'/B.'.'USE I.'.' '/  a IF < 100 LOG FILE NOT MAINTAINED
DATA LQSFIL/'TF5-L7S758+OAML00.'  a CSFS USE - DO NOT CHANGE!
DATA LQSF/1 ' /  a EXTERNAL NAME OF LOG FILE
DATA LQPKT/'MQUEUE'.4+0/  a CSFS END - DO NOT CHANGE!
END
```

**SYSS-BLOCK**

081
DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

TYPE CLRAFCN
CLASS 1110
CLASS ASH
EOF
DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

001

$NDG DAN FORTRAN PROCEDURE FORPROCS/
&POP.LF DAM.FORPROCS/..FORPROCS/
$NDG DAN FORTRAN PROCEDURE TFORM-PROCS/
&POP.LF DAM.TFORM-PROCS/..TFORM-PROCS/
$NDG DAN FORTRAN PROCEDURE PICSDEF-PROC/
&POP.LF DAM.PICDEF-PROC/..PICDEF-PROC/
$NDG DAN FORTRAN PROCEDURE MAXBYT-PROC/
&POP.LF DAM.MAXBYT-PROC/..MAXBYT-PROC/
$NDG DAN FORTRAN PROCEDURE MAXICE-PROC/
&POP.LF DAM.MAXICE-PROC/..MAXICE-PROC/
$NDG DAN FORTRAN PROCEDURE MAXINT-PROC/
&POP.LF DAM.MAXINT-PROC/..MAXINT-PROC/
$NDG DAN FORTRAN PROCEDURE NULCHR-PROC/
&POP.LF DAM.NULCHR-PROC/..NULCHR-PROC/
$NDG DAN FORTRAN PROCEDURE NULCST-PROC/
&POP.LF DAM.NULCST-PROC/..NULCST-PROC/
$NDG DAN FORTRAN PROCEDURE ASMDIF-PROC/
&POP.LF DAM.ASMDEF-PROC/..ASMDEF-PROC/
$NDG DAN FORTRAN PROCEDURE FIDEF-PROC/
&POP.LF DAM.FIDEF-PROC/..FIDEF-PROC/
$NDG DAN FORTRAN PROCEDURE FACBIT-PROC/
&POP.LF DAM.FACBIT-PROC/..FACBIT-PROC/
$NDG DAN FORTRAN PROCEDURE KOMLUS-PROC/
&POP.LF DAM.KOMLUS-PROC/..KOMLUS-PROC/
$NDG DAN FORTRAN PROCEDURE KOMIO-PROC/
&POP.LF DAM.KOMIO-PROC/..KOMIO-PROC/
$NDG DAN FORTRAN PROCEDURE PXBDEF-PROC/
&POP.LF DAM.PXBDEF-PROC/..PXBDEF-PROC/
$NDG DAN FORTRAN PROCEDURE CBDEF-PROC/
&POP.LF DAM.CBDEF-PROC/..CBDEF-PROC/
$NDG DAN FORTRAN PROCEDURE KOMSLM-PROC/
&POP.LF DAM.KOMSLM-PROC/..KOMSLM-PROC/
$NDG DAN FORTRAN PROCEDURE PRDDEF-PROC/
&POP.LF DAM.PRDDEF-PROC/..PRDDEF-PROC/
$NDG DAN FORTRAN PROCEDURE KOMIR-PROC/
&POP.LF DAM.KOMIR-PROC/..KOMIR-PROC/
$NDG DAN FORTRAN PROCEDURE KOMINET-PROC/
&POP.LF DAM.KOMNET-PROC/..KOMNET-PROC/
$NDG DAN FORTRAN PROCEDURE LSTLUB-PROC/
&POP.LF DAM.LSTLUB-PROC/..LSTLUB-PROC/
$NDG DAN FORTRAN PROCEDURE WINDOW-PROC/
&POP.LF DAMWINDOW-PROC/..WINDOW-PROC/
$NDG DAN ASSEMBLER PROCEDURE GETOPT-APROC/
&POP.L DAM.GETOPT-APROC/..GETOPT-APROC/
$NDG DAN ASSEMBLER PROCEDURE KONQAT-APROC/
&POP.L DAM.KONQAT-APROC/..KONQAT-APROC/
$NDG DAN FORTRAN PROCEDURE NERDET-PROC/
&POP.LF DAM.NERDET-PROC/..NERDET-PROC/
$NDG DAN FORTRAN PROCEDURE PODEF-PROC/
&POP.LF DAM.PODEF-PROC/..PODEF-PROC/
$NDG DAN FORTRAN PROCEDURE KOMLEN-PROC/
&POP.LF DAM.KOMLEN-PROC/..KOMLEN-PROC/
$NDG DAN FORTRAN PROCEDURE KOMKS-PROC/
&POP.LF DAM.KOMKS-PROC/..KOMKS-PROC/

K-13
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

$HOG DAM NATOHI/
$FOR.S DAM.HETONI/
$ADD DAM.SYS-FOROPT
$HOG DAM HILLBL/
$FOR.S DAM.HILLBL/
$ADD DAM.SYS-FOROPT
$HOG DAM IDEF /
$FOR.S DAM.IDEF /
$ADD DAM.SYS-FOROPT
$HOG DAM INVORI/
$FOR.S DAM.INVORI/
$ADD DAM.SYS-FOROPT
$HOG DAM INVWIN/
$FOR.S DAM.INVWIN/
$ADD DAM.SYS-FOROPT
$HOG DAM ISRTBA/
$FOR.S DAM.ISRTBA/
$ADD DAM.SYS-FOROPT
$HOG DAM ISRTBD/
$FOR.S DAM.ISRTBD/
$ADD DAM.SYS-FOROPT
$HOG DAM KSPRED/
$FOR.S DAM.KSPRED/
$ADD DAM.SYS-FOROPT
$HOG DAM LSPRED/
$FOR.S DAM.LSPRED/
$ADD DAM.SYS-FOROPT
$HOG DAM MSPRED/
$FOR.S DAM.MSPRED/
$ADD DAM.SYS-FOROPT
$HOG DAM NEOPIC/
$FOR.S DAM.NEOPIC/
$ADD DAM.SYS-FOROPT
$HOG DAM PRH1PX/
$FOR.S DAM.PRH1PX/
$ADD DAM.SYS-FOROPT
$PACK DAM.
$HOG DAM PRH2PX/
$FOR.S DAM.PRH2PX/
$ADD DAM.SYS-FOROPT
$HOG DAM QUAD/
$FOR.S DAM.QUAD/
$ADD DAM.SYS-FOROPT
$HOG DAM RITADD/
$FOR.S DAM.RITADD/
$ADD DAM.SYS-FOROPT
$HOG DAM RL21SX/
$FOR.S DAM.RL21SX/
$ADD DAM.SYS-FOROPT
$HOG DAM RL29X/
$FOR.S DAM.RL29X/
$ADD DAM.SYS-FOROPT
$HOG DAM RL45X/
$FOR.S DAM.RL45X/
$ADD DAM.SYS-FOROPT
$HOG DAM ROTHX/
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

$FOR. S DAM. ROTCNX/
$ADD. DAM. SYS. FOROPT
$HDD. DAM. ROTCOL/
$FOR. S DAM. ROTCOL/
$ADD. DAM. SYS. FOROPT
$HDD. DAM. ROTROM/
$FOR. S DAM. ROTROM/
$ADD. DAM. SYS. FOROPT
$HDD. DAM. SHFTBC/
$BASH. FS DAM. SHFTBC/
$HDD. DAM. SSSP/
$FOR. S DAM. SSSP/
$ADD. DAM. SYS. FOROPT
$HDD. DAM. SUBHI/
$FOR. S DAM. SUBHI/
$ADD. DAM. SYS. FOROPT
$HDD. DAM. VALKEY/
$FOR. S DAM. VALKEY/
$ADD. DAM. SYS. FOROPT
$HDD. DAM. VALLBL/
$FOR. S DAM. VALLBL/
$ADD. DAM. SYS. FOROPT
$HDD. DAM. WVRT/
$FOR. S DAM. WVRT /
$ADD. DAM. SYS. FOROPT
$HDD. DAM. EAPRNT/DAM
$BASH. FS DAM. EAPRNT/DAM
$HDD. DAM. EAREAD/DAM
$BASH. FS DAM. EAREAD/DAM
$HDD. DAM. EDATE/DAM
$BASH. FS DAM. EDATE/DAM
$HDD. DAM. ERFACL/DAM
$BASH. FS DAM. ERFACL/DAM
$HDD. DAM. ERFTH/DAM
$BASH. FS DAM. ERFTH/DAM
$HDD. DAM. ERPCHA/DAM
$BASH. FS DAM. ERPCHA/DAM
$Pack DAM. 
$HDD. DAM. ERPCT/DAM
$BASH. FS DAM. ERPCT/DAM
$HDD. DAM. ERPRCA/DAM
$BASH. FS DAM. ERPRCA/DAM
$HDD. DAM. EAPRNT/DAM
$BASH. FS DAM. EAPRNT/DAM
$HDD. DAM. ERPRTA/DAM
$BASH. FS DAM. ERPRTA/DAM
$HDD. DAM. ERREAD/DAM
$BASH. FS DAM. ERREAD/DAM
$HDD. DAM. ERRED/A/DAM
$BASH. FS DAM. ERRED/A/DAM
$HDD. DAM. ERSUPS/DAM
$BASH. FS DAM. ERSUPS/DAM
$HDD. DAM. ERMTAT/DAM
$BASH. FS DAM. ERMTAT/DAM
$HDD. DAM. PICPRO/
$FOR. S DAM. PICPRO/

K-17
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SHDO DAN ER1OW/DAM
SASH.FS DAN.ER1OW/DAM
SHDO DAN ERMAIT/DAM
SASH.FS DAN.ERMAIT/DAM
SHDO DAN TRUAL/
8FOR.9 DAN.TRUAL/
8ADD DAN.SYS-FOROPT
SHDO DAN TRU CST/
8FOR.9 DAN.TRUCST/
8ADD DAN.SYS-FOROPT
SHDO DAN KMO03R/
8FOR.9 DAN.KMO03R/
8ADD DAN.SYS-FOROPT
SHDO DAN QUARBN/
8FOR.9 DAN.QUARBN/
8ADD DAN.SYS-FOROPT
SHDO DAN SYSAOO/DAM
8FOR.9 DAN.SYSAOO/DAM
8ADD DAN.SYSAOO/DAM
IMOO
CAM.sys-roROPT
IMOO
CAM.KMXXOS/
8FOR.9 DAN.KMXXOS/
8ADD DAN.SYS-FOROPT
IMOO
CAM.CALSCA/
8FOR.9 DAN.CALSCA/
8ADD DAN.SYS-FOROPT
IMOO
CAM.KM000U/
8FOR.9 DAN.KM000U/
8ADD DAN.SYSAOO/DAM
SPACK DAN.
SHDO DAN PRCEXI/
8FOR.9 DAN.PRCEXI/
8ADD DAN.SYS-FOROPT
SHDO DAN P1TCALSS/
8FOR.9 DAN.P1TCALSS/
8ADD DAN.SYS-FOROPT
SHDO DAN P1TCALSS-HAP/
8FOR.9 DAN.DITCOP/
8ADD DAN.DITCOP/
8FOR.9 DAN.DITCOP/
8ADD DAN.SYS-FOROPT
SHDO DAN DITEK1/
8FOR.9 DAN.DITEK1/

K-19
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SADD DAM.SYS-FOROPT
SHDO DAM CONEXI/
SFOR.S DAM.CONEXI/
SADD DAM.SYS-FOROPT
SHDO DAM PLCGR7/
SFOR.S DAM.PLCGRT/
SADD DAM.SYS-FOROPT
SHDO DAM KNPLDO/
SFOR.S DAM.KNPLO/
SADD DAM.SYS-FOROPT
SHDO DAM LCSTEQ/
SFOR.S DAM.LCSTEQ/
SADD DAM.SYS-FOROPT
SHDO DAM PLCNP /
SFOR.S DAM.PLCPN /
SADD DAM.SYS-FOROPT
SHDO DAM PSTD /
SFOR.S DAM.PSTSD /
SADD DAM.SYS-FOROPT
SHDO DAM CCLAD /
SFOR.S DAM.CCLAD /
SADD DAM.SYS-FOROPT
SHDO DAM PICPA /
SFOR.S DAM.PICPA /
SADD DAM.SYS-FOROPT
SHDO DAM PICPA /
SFOR.S DAM.PICPA /
SADD DAM.SYS-FOROPT
SHDO DAM PIC PA /
SFOR.S DAM.PIC PA /
SADD DAM.SYS-FOROPT
SHDO DAM PICPA /
SFOR.S DAM.PICPA /
SADD DAM.SYS-FOROPT
SHDO DAM PICPA /
SFOR.S DAM.PICPA /
SADD DAM.SYS-FOROPT
SHDO DAM LOGE/
SASM.FS DAM.LOGE /
SHDO DAM NTA BS/DAM
SASM.FS DAM.NTBSP/DAM
SHDO DAM CBINIT/
SFOR.S DAM.CBINIT /
SADD DAM.SYS-FOROPT
SHDO DAM WARNS /
SFOR.S DAM.WARNS /
SADD DAM.SYS-FOROPT
SPA 'K DAM
SHDO DAM 1DUP /
SASM.FS DAM.1DUP /
SHDO DAM LENCST /
SFOR.S DAM.LEN CST /
SADD DAM.SYS-FOROPT

K-20
DOO DNN PSTART/
DOO DNN.PSTART/
GADD DNN.SYS-FOROPT
DOO DNN READS/
DOO DNN.READS/
GADD DNN.SYS-FOROPT
DOO DNN PIR-J14/
DOO DNN.PICD14/
GADD DNN.SYS-FOROPT
DOO DNN WRITE4 /
DOO DNN.WRITE4/
GADD DNN.SYS-FOROPT
DOO DNN YERA4P/
DOO DNN.YERA4P/
GADD DNN.SYS-FOROPT
DOO DNN YEP40/
DOO DNN.YEP40/
GADD DNN.SYS-FOROPT
DOO DNN VERA40/
DOO DNN.VERA40/
GADD DNN.SYS-FOROPT
DOO DNN VCA01/
DOO DNN.VCA01/
GADD DNN.SYS-FOROPT
DOO DNN KNOCLE/
DOO DNN.KNOCLE/
GADD DNN.SYS-FOROPT
DOO DNN KMDAL1/
DOO DNN.KMDAL1/
GADD DNN.SYS-FOROPT
DOO DNN KMDEN/
DOO DNN.KMDEN/
GADD DNN.SYS-FOROPT
DOO DNN KMDHEA/
DOO DNN.KMDHEA/
GADD DNN.SYS-FOROPT
DOO DNN KMDHEW/
DOO DNN.KMDHEW/
GADD DNN.SYS-FOROPT
DOO DNN KMDEXP/
DOO DNN.KMDEXP/
GADD DNN.SYS-FOROPT
DOO DNN KMDIM/
DOO DNN.KMDIM/
GADD DNN.SYS-FOROPT
DOO DNN KMDSYM/
DOO DNN.KMDSYM/
GADD DNN.SYS-FOROPT
DOO DNN KMDHCR/
DOO DNN.KMDHCR/
GADD DNN.SYS-FOROPT
DOO DNN KMDZMN/
DOO DNN.KMDZMN/
GADD DNN.SYS-FOROPT

ORIGINAL PAGE IS OF POOR QUALITY
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

PACK DAM.
&MOO DAM KM0ON/
&FOR.S DAM.KM0ON/
&ADD DAM.SYS-FOROPT
&MOO DAM KM0OFF/
&FOR.S DAM.KM0OFF/
&ADD DAM.SYS-FOROPT
&MOO DAM KM0NAM/
&FOR.S DAM.KM0NAM/
&ADD DAM.SYS-FOROPT
&MOO DAM KM0R1/
&FOR.S DAM.KM0R1/
&ADD DAM.SYS-FOROPT
&MOO DAM KM0OPT/
&FOR.S DAM.KM0OPT/
&ADD DAM.SYS-FOROPT
&MOO DAM KM0EX/
&FOR.S DAM.KM0EX/
&ADD DAM.SYS-FOROPT
&MOO DAM KM0POL/
&FOR.S DAM.KM0POL/
&ADD DAM.SYS-FOROPT
&MOO DAM KM0COL/
&FOR.S DAM.KM0COL/
&ADD DAM.SYS-FOROPT
&MOO DAM KM0R41/
&FOR.S DAM.KM0R41/
&ADD DAM.SYS-FOROPT

K-22
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

$HDO DAM 14KCOLR/
$FOR. S DAM.14KCOLR/
$ADD DAM.SYS-FOROPT
$HDO DAM QCONST/
$FOR. S DAM.QCONST/
$ADD DAM.SYS-FOROPT
$PACK DAM.
$HDO DAM KMODGO/
$FOR. S DAM.KMODGO/
$ADD DAM.SYS-FOROPT
$HDO DAM STREOG/
$FOR. S DAM.STREOG/
$ADD DAM.SYS-FOROPT
$HDO DAM KMODATT/
$FOR. S DAM.KMODATT/
$ADD DAM.SYS-FOROPT
$HDO DAM KMODPAQ/
$FOR. S DAM.KMODPAQ/
$ADD DAM.SYS-FOROPT
$HDO DAM KMODRAD/
$FOR. S DAM.KMODRAD/
$ADD DAM.SYS-FOROPT
$HDO DAM KMODTIC/
$FOR. S DAM.KMODTIC/
$ADD DAM.SYS-FOROPT
$HDO DAM ERTEWP/DAM
$ASH.FS DAM.ERTEWP/DAM
$HDO DAM TRECVR/
$FOR. S DAM.TRECVR/
$ADD DAM.SYS-FOROPT
$HDO DAM PUTHEX/
$FOR. S DAM.PUTHEX/
$ADD DAM.SYS-FOROPT
$HDO DAM CALSPA/
$FOR. S DAM.CALSPA/
$ADD DAM.SYS-FOROPT
$HDO DAM RO3BSQ/
$FOR. S DAM.RO3BSQ/
$ADD DAM.SYS-FOROPT
$HDO DAM ERSPRTC/
$ASH.FS DAM.ERSPRTC/
$ADD DAM.ERSPRTC/VIRTUAL
$ASH.FS DAM.ERTSIDC/VIRTUAL
$HDO DAM PICTAB/VIRTUAL
$ASH.FS DAM.PICTAB/VIRTUAL
$HDO DAM CONTROL/VIRTUAL
$ASH.FS DAM.CONTROL/VIRTUAL
$HDO DAM CLASSIFY/VIRTUAL
$ASH.FS DAM.CLASSIFY/VIRTUAL
$HDO DAM PRCLASS/VIRTUAL
$ASH.FS DAM.PRCLASS/VIRTUAL
$HDO DAM PLCLASS/VIRTUAL
$ASH.FS DAM.PLCLASS/VIRTUAL
$HDO DAM STATUS/VIRTUAL
$ASH.FS DAM.STATUS/VIRTUAL
$HDO DAM DITCOP/VIRTUAL

K-23
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-COMPILE 012

K-24
OAK PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

$ADD OAM.SYS-FOROPT
$ADD OAM CLROMD/
$ASH.FS OAM.CLROMD/
$ADD OAM CSTM4RL/
$ADD OAM.CSTM4RL/

$ADD OAM.SYS-FOROPT
$ADD OAM.CBS4CS/
$FOR.S OAM.CBS4CS/

$ADD OAM.SYS-FOROPT
$ADD OAM.CBS4IN/
$FOR.S OAM.CBS4IN/

$ADD OAM.SYS-FOROPT
$ADD OAM.CBST4L/
$FOR.S OAM.CBST4L/

$ADD OAM.SYS-FOROPT
$ADD OAM.CBST4L/

$ADD OAM.SYS-FOROPT
$MOO OAM.OETBYT/
$ASH.FS OAM.OETBYT/

$ADD OAM.OETCHR/
$ASH.FS OAM.OETCHR/

$ADD OAM.OETCHR/
$ASH.FS OAM.OETCHR/

$ADD OAM.OETCHR/
$ASH.FS OAM.OETCHR/

$ADD OAM.ICHR/
$SSH.0AM.ICE/

$ADD OAM.ICHR/
$SSH.0AM.ICH/

$ADD OAM.I4KONE/
$SSH.0AM.I4KONE/

$ADD OAM.I4KTWO/
$SSH.0AM.I4KTWO/

$ADD OAM.KONE41/
$SSH.0AM.KONE41/

$ADD OAM.KTWO41/
$SSH.0AM.KTWO41/

$ADD OAM.LINTEQ/
$SSH.0AM.LINTEQ/

$ADD OAM.LINTEQ/
$SSH.0AM.LINTEQ/

$SSH.0AM MOVBST/ASM
$ASH.FS OAM MOVBST/ASM

$SSH.0AM MOVBYT/
$ASH.FS OAM MOVBYT/

$SSH.0AM MOVBYT/
$ASH.FS OAM MOVBYT/

$SSH.0AM MOVCHR/
$ASH.FS OAM MOVCHR/

$PACK OAM.
$SSH.0AM MOVCCST/ASM
$ASH.FS OAM MOVCCST/ASM

$SSH.0AM MOVDBY/
$SSH.0AM MOVDBY/

K-25
SYSTEM IMPLEMENTATION

K-27
DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

@ADD DAN.SYS-FOROPT
@ADD DAN.PROEX1/
@ADD DAN.PROEX1/
@ADD DAN.SYS-FOROPT
@ADD DAN.PROL19/
@ADD DAN.SYS-FOROPT
@ADD DAN.PRCMAP/
@ADD DAN.SYS-FOROPT
@ADD DAN.PRTDET/VIRTUAL
@ADD.FS DAN.PRTDET/VIRTUAL
@ADD DAN.PRTDET-NAP/VIRTUAL
@ADD DAN.PRTDET-MAP/VIRTUAL
@ADD DAN.PRTDET-MAP/VIRTUAL
@ADD DAN.PROEX1/
@ADD DAN.PRCMAP/
@ADD DAN.PRCMAP/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
@ADD DAN.SYS-FOROPT
@ADD DAN.REALT/
DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

&FOR.S DAM.DISMS/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM CALCOL/  
&FOR.S DAM.CALCOL/  
&ADD DAM.SYS-FOROPT  
&PACK DAM.  
&HDQ DAM PICPI9/  
&FOR.S DAM.PICPI9/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM PROP19/  
&FOR.S DAM.PROP19/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM KMOTAB/  
&FOR.S DAM.KMOTAB/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM KMDF/  
&FOR.S DAM.KMDF/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM R030SK/  
&FOR.S DAM.R030SK/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM SETMOD/  
&FOR.S DAM.SETMOD/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM P1TROL/  
&FOR.S DAM.P1TROL/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM LDREC8/  
&FOR.S DAM.LDREC8/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM PICPI3/  
&FOR.S DAM.PICPI3/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM KMDF/  
&FOR.S DAM.KMDF/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM KMODCRO/  
&FOR.S DAM.KMODCRO/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM PRODET-MAP/  
&FOR.S DAM.PROP13/  
&ADD DAM.SYS-FOROPT  
&HDQ DAM DFLE-MAP/  
&HDQ DAM DFLE-MAP/VIRTUAL  
&ADD DAM SYPGET/DAM  
&ADD DAM SYPGET/DAM  
&ADD DAM ANO/  
&ADD DAM ANO/  
&ADD DAM.SYS-FOROPT  
&ADD DAM OWP/  
&ADD DAM OWP/  
&ADD DAM.SYS-FOROPT
DAM PACKAGE APPENDIX K  
SYSTEM IMPLEMENTATION  

@SHOO DAM OUV/CLARKE1966  
@SFOR.S DAM.OUV/CLARKE1966  
@ADD DAM.SYS-FOROPT  
@SHOO DAM P44/  
@SFOR.S DAM.P44/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM A4P/  
@SFOR.S DAM.A4P/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM P40/  
@SFOR.S DAM.P40/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM OUV/CLARKE1966  
@SFOR.S DAM.OUV/CLARKE1966  
@ADD DAM.SYS-FOROPT  
@SHOO DAM A4P/  
@SFOR.S DAM.A4P/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM P40/  
@SFOR.S DAM.P40/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM OUV/CLARKE1966  
@SFOR.S DAM.OUV/CLARKE1966  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PICD13/  
@SFOR.S DAM.PICD13/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PICLQ-MAP/  
@SFOR.S DAM.PICLQ-MAP/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PICEX1/  
@SFOR.S DAM.PICEX1/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PICFA3/  
@SFOR.S DAM.PICFA3/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PICFAC/  
@SFOR.S DAM.PICFAC/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PICG13/  
@SFOR.S DAM.PICG13/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM OPNO?N/  
@SFOR.S DAM.OPNO?N/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PICR000/  
@SFOR.S DAM.PICR000/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM OPDL13/  
@SFOR.S DAM.OPDL13/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PRO000/  
@SFOR.S DAM.PRO000/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM IDCPRO/  
@SFOR.S DAM.IDCPRO/  
@ADD DAM.SYS-FOROPT  
@SHOO DAM PRO000/  
@SFOR.S DAM.PRO000/  

K-30
DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SADD DAM.SYS-ROOPT
SMDO DAM CLSHDO/
SFOR.S DAM CLSHDO/
SADD DAM.SYS-ROOPT
SMDO DAM OPEN3/
SFOR.S DAM.OPEN3/
SADD DAM.SYS-ROOPT
SMDO DAM OPN3EN/
SFOR.S DAM.OPN3EN/
SADD DAM.SYS-ROOPT
SMDO DAM OP3DSK/
SFOR.S DAM.OP3DSK/
SADD DAM.SYS-ROOPT
SMDO DAM PXBOMP/
SFOR.S DAM.PXBOMP/
SADD DAM.SYS-FOROPT
SMDO DAM R03BIP/
SFOR.S DAM.R03BIP/
SADD DAM.SYS-ROOPT
SMDO DAM SM0CHA/
SFOR.S DAM.SM0CHA/
SADD DAM.SYS-ROOPT
SMDO DAM DETCHA/
SFOR.S DAM.DETCHA/
SADD DAM.SYS-ROOPT
SADD DAM PACK
SMDO DAM CALWIN/
SFOR.S DAM.CALWIN/
SADD DAM.SYS-ROOPT
SMDO DAM CLOSES/
SFOR.S DAM.CLOSES/
SADD DAM.SYS-ROOPT
SMDO DAM CL3BIP/
SFOR.S DAM.CL3BIP/
SADD DAM.SYS-ROOPT
SMDO DAM CROPOH/
SFOR.S DAM.CROPOH/
SADD DAM.SYS-ROOPT
SMDO DAM DSSPR/
SFOR.S DAM.DSSPR/
SADD DAM.SYS-ROOPT
SMDO DAM GENTIC/
SFOR.S DAM.GENTIC/
SADD DAM.SYS-ROOPT
SMDO DAM GETDSR/
SADD DAM.SYS-ROOPT
SADD DAM.IDLU3/
SFOR.S DAM.IDLU3/
SADD DAM.SYS-ROOPT
SMDO DAM T3RTMS/
SADD DAM.SYS-ROOPT
SMDO DAM T3RTSA/
SADD DAM.SYS-ROOPT
SADD DAM.SYS-ROOPT
DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

$NODE DAN.ISRTMA/
$FOR.S DAN.ISRTMA/
$ADD DAN.SYS-FOROPT
$NODE DAN.ISRTMD/
$FOR.S DAN.ISRTMD/
$ADD DAN.SYS-FOROPT
$NODE DAN.ISRTSD/
$FOR.S DAN.ISRTSD/
$ADD DAN.SYS-FOROPT
$NODE DAN.HATPRT/
$FOR.S DAN.HATPRT/
$ADD DAN.SYS-FOROPT
$NODE DAN.MVCNT/
$BASH.FB DAN.MVCNT/
$NODE DAN.NVIATO/
$BASH.FB DAN.NVIATO/
$NODE DAN.OPENPR/
$FOR.S DAN.OPENPR/
$ADD DAN.SYS-FOROPT
$NODE DAN.OP3TAP/
$FOR.S DAN.OP3TAP/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ANCL/
$FOR.S DAN.03ANCL/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ANOT/
$FOR.S DAN.03ANOT/
$ADD DAN.SYS-FOROPT
$NODE DAN.03HDR/
$FOR.S DAN.03HDR/
$ADD DAN.SYS-FOROPT
$PACK DAN.
$NODE DAN.03SAR/
$FOR.S DAN.03SAR/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZAK/
$FOR.S DAN.03ZAK/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZAM/
$FOR.S DAN.03ZAM/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZPM/
$FOR.S DAN.03ZPM/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZPR/
$FOR.S DAN.03ZPR/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZPR/
$FOR.S DAN.03ZPR/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZPR/
$FOR.S DAN.03ZPR/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZPR/
$FOR.S DAN.03ZPR/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZPR/
$FOR.S DAN.03ZPR/
$ADD DAN.SYS-FOROPT
$NODE DAN.03ZPR/
$PACK DAN.
$NODE DAN.KMDREN/
$FOR.S DAN.KMDREN/
$ADD DAN.SYS-FOROPT
$NODE DAN.KMOWIN/
$FOR.S DAN.KMOWIN/
$ADD DAN.SYS-FOROPT

K-32
DAN PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

@FOR.S DAM.PRINC/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM DEG/ 
@FOR.S DAM.DEG/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM DEDNS/ 
@FOR.S DAM.DEDNS/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM SPLIT/ 
@FOR.S DAM.SPLIT/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM CLADET/ 
@FOR.S DAM.CLADET/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM HAPRNT/ 
@FOR.S DAM.HAPRNT/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM IDFILE/ 
@FOR.S DAM.IDFILE/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM LOCATE/ 
@FOR.S DAM.LOCATE/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM MP3HOP/ 
@FOR.S DAM.MP3HOP/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM CONGIA/ 
@FOR.S DAM.CONGIA/ 
@ADD DAM.SYS-FOROPT 
@HOD DAM PRTCLASS-MAP/ 
@HOD DAM STATUS/ 
@FOR.S DAM.STATUS/ 
@ADD DAM.SYS-FOROPT

SYS-COMPILE
624
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYSCALL SET

-MSO.N SYS-COLLECT RUNSTREAM FOR SNAP PROCESSOR
-MSO.N ----------------------------------------
-MSO.N •MSO.N •MSO.N HISTORY
-MSO.N -------
-MSO.N E H SCHLOSSER LEC 03/03/73 ORIGINAL CODE
-MSO.N E H SCHLOSSER LEC 07/31/79 ADD F OPTION (QUARTER-WORD MODE)
-MSO.N E H SCHLOSSER LEMSCO 09/18/80 CHANGE PRTDENS TO PRTDET
-MSO.N -------
-MSO.N •MSO.N •MSO.N EXCEPTIONS
-MSO.N -------
-MSO.N 
-MSO.N 1. THIS ELEMENT MUST NOT BE ADD-ED FROM THE DAM PROGRAM FILE.
-MSO.N SINCE IT &PACK-S THAT FILE. INSTEAD IT MUST BE COPIED TO
-MSO.N TPFS. AND ADD-ED FROM THERE.
-MSO.N -------
-MSO.N 
-MSO.N ADD DAM PACKAGE -- COMPILE SYS-BLOCK AND COLLECT MAIN PROGRAMS
-MSO.N FOR.S DAM.SYS-BLOCK
-MSO.N ADD DAM.SYS-FOROPT
-MSO.N &PACK.SR DAM. CAUTION: ADD ELEMENT MUST NOT PACK ITS OWN FILE!
-MSO.N &PREP DAM.
-MSO.N &MAP.FLZ DAM.ERSPRTRAN-MAP.DAM.ERSPRTRAN NEVER VIRTUAL!!
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.IDFILE-MAP.DAM.IDFILE NEVER VIRTUAL!!
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.LOCATE-MAP.DAM.LOCATE NEVER VIRTUAL!!
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.USMAP-MAP.DAM.USMAP NEVER VIRTUAL!!
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.ERTSIDC-MAP/VIRTUAL.DAM.ERTSIDC
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.PICTAB-MAP.DAM.PICTAB
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.CONTROL-MAP.DAM.CONTROL
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.CLASSIFY-MAP/VIRTUAL.DAM.CLASSIFY
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.PRTDET-MAP/VIRTUAL.DAM.PRTDET
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.PRTCLASS-MAP/VIRTUAL.DAM.PRTCLASS
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.PTCLASS-MAP/VIRTUAL.DAM.PTCLASS
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.STATUS-MAP.DAM.STATUS
-MSO.N ADD DAM.SYS-HAOPPT
-MSO.N &MAP.FLZ DAM.DITCOP-MAP/VIRTUAL.DAM.DITCOP
-MSO.N ADD DAM.SYS-HAOPPT

K-37
<table>
<thead>
<tr>
<th>DAM PACKAGE APPENDIX K</th>
<th>SYSTEM IMPLEMENTATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>8DELETE.S DAM.ERPRCH/DAM</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.OLSTSQ/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.OIAERP/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.GAPFOC/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.GAPLUF/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.GAPLUR/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.FOCUS/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.HULSUB/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.OMPMTIC/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PICPR3/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.ERERR/DAM</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.EREXIT/DAM</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PICDIS/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PICPAB/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.99450/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PCL15/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PICPA9/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PICPRG/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PICTO3/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PICTO9/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.CLADE5/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.CLADE9/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.OPRCLA/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.SLMCLA/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.COPYPX/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PIC129/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PIC345/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PIC878/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.CLA129/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.CLA345/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.H2TON1/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.HILLBL/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.10ERT/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.INVOR1/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.INVWIN/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.ISRTBA/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.ISRTBD/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.KSPRED/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.LSREP/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.MSREP/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.NECROIC/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PRIPX/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.PRM2PX/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.QUAD/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.RITAAD/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.RL215X/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.RLEGX/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.ROTCMX/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.ROTCOL/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.ROTROM/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.SHFTBC/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.SSFR/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.SUB1/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.VALKEY/</td>
<td></td>
</tr>
<tr>
<td>8DELETE.S DAM.VALLBL/</td>
<td></td>
</tr>
</tbody>
</table>
TOD PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

DELETE.S DAM.HRVERT/
DELETE.S DAM.EAPRNT/DAM
DELETE.S DAM.EAREAD/DAM
DELETE.S DAM.ERDATE/DAM
DELETE.S DAM.ERFACL/DAM
DELETE.S DAM.ERFITH/DAM
DELETE.S DAM.ERPCHA/DAM
DELETE.S DAM.ERPCT/DAM
DELETE.S DAM.ERPRA/DAM
DELETE.S DAM.EAPRNT/DAM
DELETE.S DAM.EPRTA/DAM
DELETE.S DAM.EREAD/DAM
DELETE.S DAM.EREDA/DAM
DELETE.S DAM.ERSUPS/DAM
DELETE.S DAM.ERTWAT/DAM
DELETE.S DAM.PICTPRO/
DELETE.S DAM.PICTOT/
DELETE.S DAM.KMDOAS/
DELETE.S DAM.KMDOFR/
DELETE.S DAM.KWRIIN/
DELETE.S DAM.ERPCS/DAM
DELETE.S DAM.ERPRCA/DAM
DELETE.S DAM.ERPRNT/DAM
DELETE.S DAM.ERPRTA/DAM
DELETE.S DAM.ERREAD/DAM
DELETE.S DAM.KMDOAD/
DELETE.S DAM.KMDOLO/
DELETE.S DAM.LENPA/
DELETE.S DAM.LOWCST/
DELETE.S DAM.SREADS/
DELETE.S DAM.SPANS/
DELETE.S DAM.GETOKH/
DELETE.S DAM.NEXTOK/
DELETE.S DAM.GETS/
DELETE.S DAM.ERIO/DAM
DELETE.S DAM.ERIOW/DAM
DELETE.S DAM.ERWAIT/DAM
DELETE.S DAM.TRUAL/
DELETE.S DAM.TRUCST/
DELETE.S DAM.KMDO9R/
DELETE.S DAM.QUARTN/
DELETE.S DAM.QUART/
DELETE.S DAM.VARSQ/
DELETE.S DAM.VARSQN/
DELETE.S DAM.SYSA00/DAM
DELETE.S DAM.KMXOS/
DELETE.S DAM.CALSCA/
DELETE.S DAM.KMDOAU/
DELETE.S DAM.DITXQT/
DELETE.S DAM.SUBWIN/
DELETE.S DAM.PRCExI/
DELETE.S DAM.PLCCLASS/
DELETE.S DAM.OITCOP/
DELETE.S DAM.OITEXI/
DELETE.S DAM.<XEI/
DELETE.S DAM.KMDPLO/

ORI age PAGE 47 OF POOR QUALITY

K-39
SYS-DELETE

- GOELETE.S OAM.LCSTEG/
- GOELETE.S OAM.PLCHAP/
- GOELETE.S OAM.PSTTP/
- GOELETE.S OAM.CLARAD/
- GOELETE.S OAM.PICPA4/
- GOELETE.S OAM.PICPA3/
- GOELETE.S OAM.PICPAR/
- GOELETE.S OAM.PICL14/
- GOELETE.S OAM.PICD19/
- GOELETE.S OAM.PICL19/
- GOELETE.S OAM.LOO2/
- GOELETE.S OAM.HNABS/DAM
- GOELETE.S OAM.L002/
- GOELETE.S OAM.NTABS/DAM
- GOELETE.S OAM.CBINIT/
- GOELETE.S OAM.WARN5/
- GOELETE.S OAM.10UP/
- GOELETE.S OAM.LENCST/
- GOELETE.S OAM.PSTART/
- GOELETE.S OAM.READS/
- GOELETE.S OAM.PICQ14/
- GOELETE.S OAM.WRITE4/
- GOELETE.S OAM.VER4P/
- GOELETE.S OAM.VER4N/
- GOELETE.S OAM.VER4U/
- GOELETE.S OAM.ARORET/
- GOELETE.S OAM.CLOSE4/
- GOELETE.S OAM.KMDCLE/
- GOELETE.S OAM.KMDAL1/
- GOELETE.S OAM.KMODEN/
- GOELETE.S OAM.KMDEEA/
- GOELETE.S OAM.KMNEW/
- GOELETE.S OAM.KMEDXP/
- GOELETE.S OAM.KMOTIM/
- GOELETE.S OAM.KMDSYM/
- GOELETE.S OAM.KMODER/
- GOELETE.S OAM.KMIZON/
- GOELETE.S OAM.KM000/
- GOELETE.S OAM.KM00FF/
- GOELETE.S OAM.KM00AN/
- GOELETE.S OAM.KM00RI/
- GOELETE.S OAM.CST41N/
- GOELETE.S OAM.CB41N/
- GOELETE.S OAM.KM00CP/
- GOELETE.S OAM.KMODG/
- GOELETE.S OAM.KM00G/
- GOELETE.S OAM.KMOPR1/
- GOELETE.S OAM.KMMXXX/
- GOELETE.S OAM.KMXXED/
- GOELETE.S OAM.XRE077/
- GOELETE.S OAM.ARE077/
- GOELETE.S OAM.KMODPN/
- GOELETE.S OAM.KM00L/
- GOELETE.S OAM.KM00L/
- GOELETE.S OAM.KM001/
- GOELETE.S OAM.KM001/
- GOELETE.S OAM.OCONST/

K-40
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>@DELETE.S DAM.KMDGOO/</td>
<td>systemctl delete</td>
</tr>
</tbody>
</table>
#DELETE.S DAM.READ3/
#DELETE.S DAM.RD3BIL/
#DELETE.S DAM.R3TREC/
#DELETE.S DAM.CLOS3R/
#DELETE.S DAM.PRSYML/
#DELETE.S DAM.NITMDG/
#DELETE.S DAM.MAPHDO/
#DELETE.S DAM.KNDPEE/
#DELETE.S DAM.KHDP0K/
#DELETE.S DAM.EBCST/
#DELETE.S DAM.PICPIE/
#DELETE.S DAM.PROPIC/
#DELETE.S DAM.DISH3S/
#DELETE.S DAM.CALC0L/
#DELETE.S DAM.PICPI9/
#DELETE.S DAM.PRDPI9/
#DELETE.S DAM.KHDJ1S/
#DELETE.S DAM.KDDT1F/
#DELETE.S DAM.RO3S3K/
#DELETE.S DAM.SETHOD/
#DELETE.S DAM.PITROL/
#DELETE.S DAM.LORE08/
#DELETE.S DAM.PICPI3/
#DELETE.S DAM.KMOINT/
#DELETE.S DAM.KMOCR0/
#DELETE.S DAM.PROPI3/
#DELETE.S DAM.SYMOET/DAM
#DELETE.S DAM.4V0/
#DELETE.S DAM.4V4/
#DELETE.S DAM.4V4/CLRKE1868
#DELETE.S DAM.4V4/
#DELETE.S DAM.4V4/
#DELETE.S DAM.NO/CLRKE1868
#DELETE.S DAM.4V4/CLRKE1868
#DELETE.S DAM.PICD13/
#DELETE.S DAM.PICEY1/
#DELETE.S DAM.PICFA3/
#DELETE.S DAM.PICFAC/
#DELETE.S DAM.PICRF/
#DELETE.S DAM.CNO00/
#DELETE.S DAM.OPH02N/
#DELETE.S DAM.PROD00/
#DELETE.S DAM.PR0L15/
#DELETE.S DAM.PR0L13/
#DELETE.S DAM.10CCRO/
#DELETE.S DAM.PRC000/
#DELETE.S DAM.CLSQ00/
#DELETE.S DAM.OPN02/
#DELETE.S DAM.OPN12/
#DELETE.S DAM.OP30SK/
#DELETE.S DAM.PX80MP/
#DELETE.S DAM.RO381P/
#DELETE.S DAM.KMDCHA/
#DELETE.S DAM.DETCHA/

K-43
DAM PACKAGE APPENDIX K
SYSTEM IMPLEMENTATION

SYS-DELETE
007

0DELETE.S DAM.CALMIN/
0DELETE.S DAM.CLOSES/
0DELETE.S DAM.CL38IP/
0DELETE.S DAM.CROPOM/
0DELETE.S DAM.OSSPR/
0DELETE.S DAM.OEBIC/
0DELETE.S DAM.OETOSR/
0DELETE.S DAM.10UL3/
0DELETE.S DAM.TSRTHS/
0DELETE.S DAM.ISRTSA/
0DELETE.S DAM.ISRTHA/
0DELETE.S DAM.ISRTHD/
0DELETE.S DAM.ISRTSD/
0DELETE.S DAM.1ATPRT/
0DELETE.S DAM.MVCON/
0DELETE.S DAM.VISTIO/
0DELETE.S DAM.OPENPR/
0DELETE.S DAM.OP3TAP/
0DELETE.S DAM.03ANCL/
0DELETE.S DAM.03ANOT/
0DELETE.S DAM.03HOR/
0DELETE.S DAM.03ZAR/
0DELETE.S DAM.03ZAH/
0DELETE.S DAM.03ZPH/
0DELETE.S DAM.03ZPR/
0DELETE.S DAM.03TR/
0DELETE.S DAM.CALCHA/
0DELETE.S DAM.KDREN/
0DELETE.S DAM.KMDIM/
0DELETE.S DAM.KMODCA/
0DELETE.S DAM.KMOSHA/
0DELETE.S DAM.KMOTOL/
0DELETE.S DAM.ERINFO/DAM
0DELETE.S DAM.USHAP/
0DELETE.S DAM.ERT510C/
0DELETE.S DAM.PICTAB/
0DELETE.S DAM.PIC000/
0DELETE.S DAM.PICFA9/
0DELETE.S DAM.PICL13/
0DELETE.S DAM.PICXQT/
0DELETE.S DAM.10CPIC/
0DELETE.S DAM.OPRPIC/
0DELETE.S DAM.CONTROL/
0DELETE.S DAM.CNXQT/
0DELETE.S DAM.MSKPIX/
0DELETE.S DAM.CLAXQT/
0DELETE.S DAM.PRTDET/
0DELETE.S DAM.CLASSIFY/
0DELETE.S DAM.PRODI9/
0DELETE.S DAM.PRODI3/
0DELETE.S DAM.PROXOT/
0DELETE.S DAM.OPRPRO/
0DELETE.S DAM.PRTCLASS/
0DELETE.S DAM.PRCXQT/
0DELETE.S DAM.RESPRC/
0DELETE.S DAM.PLCC000/
OAM PACKAGF APPENDIX K
SYSTEM IMPLEMENTATION

DELETES OAM.DITDUP/
DELETES OAM.DITDUP/
DELETES OAM.DOSCNT/
DELETES OAM.DORECP/
DELETES OAM.DOSORT/
DELETES OAM.DPMMIN/
DELETES OAM.099PR3/
DELETES OAM.ERESTD/
DELETES OAM.EVOR1/
DELETES OAM.EVWIN/
DELETES OAM.FCPRRT/
DELETES OAM.FLNP0/
DELETES OAM.FCERT/
DELETES OAM.OCHOM/
DELETES OAM.0EPRAD/
DELETES OAM.MDUNIT/
DELETES OAM.IDERTS/
DELETES OAM.JOIN2N/
DELETES OAM.OP301P/
DELETES OAM.PROVF1/
DELETES OAM.PRCHR/
DELETES OAM.PX4AP/
DELETES OAM.PX4AP/
DELETES OAM.PX4PM/
DELETES OAM.PX4PR/
DELETES OAM.SHASAM/
DELETES OAM.SHAP3/
DELETES OAM.PRTMQO/
DELETES OAM.PRTINC/
DELETES OAM.0EG/
DELETES OAM.0EOMS/
DELETES OAM.SPLIT/
DELETES OAM.CLADET/
DELETES OAM.MAPRT/
DELETES OAM.LOFILE/
DELETES OAM.LOCATE/
DELETES OAM.OP3MDP/
DELETES OAM.CONDIA/
DELETES OAM.STATUS/
**DAM PACKAGE APPENDIX K**

**SYSTEM IMPLEMENTATION**

**BMSO.N**

**SYN-GENCON RUNSTREAM TO GENERATE SYS-COMPILE EXEC COMMAND RUNSTREAM**

**BMSO.N**

**HISTORY**

**BMSO.N**

**BMSO.N**

**BMSO.N**

**BMSO.N**

**BMSO.N**

**BMSO.N**

**BMSO.N**

**BMSO.N**

**NYGO DAM PACKAGE SYSTEM GENERATION CONTROL STREAMS (SYN-GENCON)**

**BMSO.KOP**

**DAM./ELM....PCF/1.DAM.**

**SKEI**

*ELT.ID DAM.SYS-COMPIL*

**INCREMENT NOUT TO (ELM)**

*IF (ELM.NOUT.3.11 = +2*

**MOD DAM ASSEMBLER PROCEDURE (ELM.NOUT.1.11/ELM.NOUT.2.11**

**PDP.L DAM (ELM.NOUT.1.11/ELM.NOUT.2.11/ELM.NOUT.1.11/ELM.NOUT.2.11**

**END**

*IF (ELM.NOUT.3.11 = +3*

**MOD DAM COBOL PROCEDURE (ELM.NOUT.1.11/ELM.NOUT.2.11**

**PDP.LC DAM (ELM.NOUT.1.11/ELM.NOUT.2.11/ELM.NOUT.1.11/ELM.NOUT.2.11**

**END**

*IF (ELM.NOUT.3.11 = +4*

**MOD DAM FORTRAN PROCEDURE (ELM.NOUT.1.11/ELM.NOUT.2.11**

**PDP.LF DAM (ELM.NOUT.1.11/ELM.NOUT.2.11/ELM.NOUT.1.11/ELM.NOUT.2.11**

**END**

*LOOP*

**PACK.SR DAM.**

**CLEAR NREL**

*INCREMENT NOUT TO (ELM)**

*IF (ELM.NOUT.3.11 = +1*

*TYPE 1 = SYMBOLIC*

*IF (ELM.NOUT.3.21 = +1*

**MOD DAM (ELM.NOUT.1.11/ELM.NOUT.2.11**

*IF (ELM.NOUT.3.21 = +2*

**SET NREL TO NREL+1**

**ASM.FS DAM ((ELM.NOUT.1.11/ELM.NOUT.2.11**

**END**

*IF (ELM.NOUT.3.21 = +3*

**SET NREL TO NREL+1**

**COB.LDIR DAM (ELM.NOUT.1.11/ELM.NOUT.2.11**

**END**

*IF (ELM.NOUT.3.21 = +4*

**ADD DAM.SYS-FOROPT**

**END**

_K-46_
IF *HREL > +20
  CLEAR HREL
  PACK DAM.
END

END

LOOP

*ELT.10 DAM.SYS-DELETE
*INCREMENT NOUT TO [ELM1] , **** DELETE SOURCE SYMBOLIC ELEMENTS ****
  IF [ELM.NOUT.3.1] = +1
  IF [ELM.NOUT.3.21] < +6
    DELETE.S DAM,[ELM.NOUT.1.11/ELM.NOUT.2.11]
  END
  END

END

LOOP

EOF

EOF

SCYCLE DAM.SYS-compile.1
SED.U DAM.SYS-compile

C / //A

EXIT

SCYCLE DAM.SYS-DELETE.1
SED.U DAM.SYS-DELETE

C / //A

EXIT
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  PRODE
PRC  PROCLAS
STA  STATUS
DI1  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)
'B'  PHASE 1.5  (ADDITIONAL PRE-PROCESSING)
'B', '6', 'G'  PHASE 3.4.9  (MAJOR PROCESSING -- FORTRAN I/O NOT ALLOWED)
'B'  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 '189' AND '245' 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 NVIA0, 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 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 NVIA0 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 NVIA0, NAMING THE '000' MONITOR AS THE
'VIA' routine and MULSUB as the 'TO' routine. The 'OOD' monitor will then read the name of the next command and call the appropriate phase 0 command routine.
MAIN PROGRAMS/Routines

APPENDIX-L

BPR.T SC  OAM.PREFACE-L  (0009)
BPR.T SC  OAM.APPENDIX-L
MSO.N
BPR.T SC  OAM.SETUP  SET UP TPFS AND TTY, AND PRINT NEWS
MSO.N
BPR.T SC  OAM.SETUP-HIA  HIERARCHY
MSO.N
BPR.T SC  OAM.SETUP/N  SET UP TPFS AND TTY, AND DON'T PRINT NEWS
MSO.N
BPR.T SC  OAM.ERSPRTCN  SUBMIT PRINT CONTROL SPECIFICATIONS
BPR.T SC  OAM.ERSPRTCN-MAP  COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPR.T SC  OAM.ERSPRTCN-MAP/VIRTUAL  COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
MSO.N
BPR.T SC  OAM.DATA/CHECKOUT  PSEUDO EXEC COMMAND: INITIATE DATA/CHECKOUT MODE
BPR.T SC  OAM.DATA-MAP  COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPR.T SC  OAM.DATA/MAP/VIRTUAL  COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
BPR.T SC  OAM.DATA-CHECK  DATA/CHECKOUT
MSO.N
BPR.T SC  OAM.IDFILE  PSEUDO EXEC COMMAND: IDENTIFY TAPE OR DISK FILE
BPR.T SC  OAM.IDFILE-MAP  COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPR.T SC  OAM.IDFILE-MAP/VIRTUAL  COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
MSO.N
BPR.T SC  OAM.LOCATE  PSEUDO EXEC COMMAND: POSITION MULTI-FILE TAPE
BPR.T SC  OAM.LOCATE-MAP  COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPR.T SC  OAM.LOCATE-MAP/VIRTUAL  COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
MSO.N
BPR.T SC  OAM.USWAP  PSEUDO EXEC COMMAND: SWAP TAPE DRIVE UNITS
BPR.T SC  OAM.USWAP-MAP  COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPR.T SC  OAM.USWAP-MAP/VIRTUAL  COLLECTOR SYMBOLICS FOR REAL ABS (NO VIRTUAL)
BPR.T SC  OAM.ERTS-DUP  DUPLICATE ERTS MSS TAPE
MSO.N
BPR.T SC  OAM.ERTSIOC  PRINT ID/ANNOTATION DATA FROM ERTS MSS TAPES
BPR.T SC  OAM.ERTSIOC-HIA  HIERARCHY
BPR.T SC  OAM.ERTSIOC/VIRTUAL  MAP OAM.ERTSIOC TO TPFS AND EXGT.1
BPR.T SC  OAM.ERTSIOC-MAP  COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPR.T SC  OAM.ERTSIOC-MAP/VIRTUAL  COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
MSO.N
BPR.T SC  OAM.PICTAB  DISPLAY/TABULATE/FACTOR/PARTITION ERTS MSS DATA
BPR.T SC  OAM.PICTAB-HIA  HIERARCHY
BPR.T SC  OAM.PICTAB/VIRTUAL  MAP OAM.PICTAB TO TPFS AND EXGT.1
BPR.T SC  OAM.PICTAB-MAP  COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
BPR.T SC  OAM.PICTAB-MAP/VIRTUAL  COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
BPR.T SC  OAM.PICO00  CALL PHASE 0 (COMMAND) ROUTINES FOR PICTAB
BPR.T SC  OAM.PICO12  CALL PHASE 1/2/9 ROUTINES FOR PICTAB
BPR.T SC  OAM.PICO345  CALL PHASE 3/4/5 ROUTINES FOR PICTAB
BPR.T SC  OAM.PICO378  CALL PHASE 6/7/8 ROUTINES FOR PICTAB
BPR.T SC  OAM.PICO15  DISPLAY MSS-DERIVED DATA (PHASE 0)
BPR.T SC  OAM.PICO13  DISPLAY RADIANCE (PHASE 3)
BPR.T SC  OAM.PICO14  DISPLAY GRADIENT/LAPLACION/VARIANCE (PHASE 4)
BPR.T SC  OAM.PICO15  DISPLAY CLASS (PHASE 5)
BPR.T SC  OAM.PICO19  DISPLAY MSS-DERIVED DATA (PHASE 9)
BPR.T SC  OAM.DISHIS  HISTOGRAM DISPLAYED DATA (PHASE 0)
BPR.T SC  OAM.PICE1  TERMINATION ROUTINE (PHASE 0)
BPR.T SC  OAM.PICFAC  FACTOR MSS CHANNELS (PHASE 0)
BPR.T SC  OAM.PICFAC3  FACTOR MSS CHANNELS (PHASE 3)
BPR.T SC  OAM.PICFAC9  FACTOR MSS CHANNELS (PHASE 9)
APPENDIX-L

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

APRT.SC DAN.PICL1S
APRT.SC DAN.PICL13
APRT.SC DAN.PICL14
APRT.SC DAN.PICL15
APRT.SC DAN.PICL19
APRT.SC DAN.PICPAR
APRT.SC DAN.PICPA2
APRT.SC DAN.PICPA4
APRT.SC DAN.PICPA6
APRT.SC DAN.PICPA8
APRT.SC DAN.PICPA9
APRT.SC DAN.PICPIC
APRT.SC DAN.PICP13
APRT.SC DAN.PICP14
APRT.SC DAN.PICP15
APRT.SC DAN.PICP19
APRT.SC DAN.PICPR0
APRT.SC DAN.PICPR3
APRT.SC DAN.PICPR9
APRT.SC DAN.PICT0T
APRT.SC DAN.PICT01
APRT.SC DAN.PICT03
APRT.SC DAN.PICT09
APRT.SC DAN.PICXQT
APRT.SC DAN.IDCPIC
APRT.SC DAN.OPRPCIC

MSO.N DAN.CONTROL
APRT.SC DAN.CONTROL-MIA
APRT.SC DAN.CONTROL/VIRTUAL
APRT.SC DAN.CONTROL-MAP
APRT.SC DAN.CONTROL-MAP/VIRTUAL
APRT.SC DAN.COMCON
APRT.SC DAN.COMADJ
APRT.SC DAN.COMDIA
APRT.SC DAN.CONEX
APRT.SC DAN.COMXQT
APRT.SC DAN.DIAERR
APRT.SC DAN.DLSSTSQ

MSO.N DAN.CLASSIFY
APRT.SC DAN.CLASSIFY-MIA
APRT.SC DAN.CLASSIFY/VIRTUAL
APRT.SC DAN.CLASSIFY-MAP
APRT.SC DAN.CLASSIFY-MAP/VIRTUAL
APRT.SC DAN.CLA00
APRT.SC DAN.CLA029
APRT.SC DAN.CLA345
APRT.SC DAN.CLADE3
APRT.SC DAN.CLADE4
APRT.SC DAN.CLADE5
APRT.SC DAN.CLADE9
APRT.SC DAN.CLAEX1
APRT.SC DAN.CLANAD
APRT.SC DAN.CLAQXT
APRT.SC DAN.CLSO2N

LIST MSS-DERIVED DATA (PHASE 0)
LIST RADIANCE (PHASE 3)
LIST GRADIENT (PHASE 4)
LIST CLASS (PHASE 5)
LIST MSS-DERIVED DATA (PHASE 9)
PARTITION FACTOR SPACE (PHASE 0)
PARTITION BY DENSITY (PHASE 3)
PARTITION BY GRADIENT/LAPLACIAN/VARIANCE (PHASE 9)
PARTITION FACTOR SPACE (PHASE 6)
PARTITION FACTOR SPACE (PHASE 9)
PICTURE MSS-DERIVED DATA (PHASE 0)
PICTURE RADIANCE (PHASE 3)
PICTURE MSS-DERIVED DATA (PHASE 9)
PROFILE MSS-DERIVED DATA (PHASE 0)
PROFILE MSS-DERIVED DATA (PHASE 3)
PROFILE MSS-DERIVED DATA (PHASE 9)
ROTATE FACTOR STRUCTURE/COEFFICIENTS
TOTAL TABULATIONS
TOTAL TABULATIONS
INITIALIZATION ROUTINE (PHASE 0)
IDENTIFY CURRENT COMMAND SPECS (UTILITY)
OPEN ALTERNATE PRINT FILES (UTILITY)
ADJUST/DIAGRAM CONTROL NETWORK
HIERSARCHY
MAP DAN.CONTROL TO TPFS AND XQ0T.1
COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
CALL PHASE 0 (COMMAND) ROUTINES FOR CONTROL
ADJUST NETWORK (PHASE 0)
DIAGRAM NETWORK (PHASE 0)
TERMINATION ROUTINE (PHASE 0)
INITIALIZATION ROUTINE (PHASE 0)
DIAGRAM ERRORS
LEAST SQUARES BI-LINEAR FIT
CLASSIFY DATA ON ERAS MSS TAPE
HIERSARCHY
MAP DAN.CLASSIFY TO TPFS AND XQ0T.1
COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
CALL PHASE 0 (COMMAND) ROUTINES FOR CLASIFY
CALL PHASE 1/2/9 ROUTINES FOR CLASIFY
CALL PHASE 3/4/5 ROUTINES FOR CLASIFY
GENERATE DETECTION FILE (PHASE 0)
GENERATE RADIANCE DETECTION FILE (PHASE 3)
GENERATE DENSITY DETECTION FILE (PHASE 4)
GENERATE CLASS DETECTION FILE (PHASE 5)
GENERATE DETECTION FILE (PHASE 5)
GENERATE DETECTION FILE (PHASE 9)
TERMINATION ROUTINE (PHASE 0)
GET/CHECK RADIANCE LIMITS (PHASE 0)
INITIALIZATION ROUTINE (PHASE 0)
CLOSE OUTPUT DETECTION FILE (UTILITY)
APPENDIX L
MAIN PROGRAMS/Routines

OPEN OUTPUT DETECTION FILE (UTILITY)
OPEN ALTERNATE PRINT FILE (UTILITY)
PLOT SPECTRAL LIMITS (UTILITY)

DISPLAY DATA FROM DETECTION FILE(S)
HIERARCHY

MAP DAM.PRTDET TO TPFS AND BXQT.1
COLLECTOR SYMBOLS FOR REAL ABSOLUTE

CALL PHASE 0 (COMMAND) ROUTINES FOR PRTDET

DISPLAY DETECTION DATA (PHASE 0)
DISPLAY DETECTION DATA (PHASE 3)
DISPLAY DETECTION DATA (PHASE 9)

TERMINATION ROUTINE

LIST DETECTION DATA (PHASE 0)
LIST DETECTION DATA (PHASE 3)
LIST DETECTION DATA (PHASE 9)

INITIALIZATION ROUTINE

IDENTIFY CURRENT COMMAND SPECs (UTILITY)
OPEN ALTERNATE PRINT FILES (UTILITY)

OUTPUT CLASSIFIED ERTS MAPS ON LINE PRINTER
HIERARCHY

MAP DAM.PRTCLASS TO TPFS AND BXQT.1
COLLECTOR SYMBOLS FOR REAL ABSOLUTE

CALL PHASE 0 (COMMAND) ROUTINES FOR PRTCLASS

MAP RADIANCE/DENSITY/CLASS (PHASE 0)
PRINT MAPS

PROVIDE UNIT HEADING (UTILITY)
RESAMPLE DETECTION PIXELS (UTILITY)

OUTPUT CLASSIFIED ERTS MAPS ON PEN PLOTTER

MAP DAM.PLTCCLASS TO TPFS AND BXQT.1
COLLECTOR SYMBOLS FOR REAL ABSOLUTE

CALL PHASE 0 (COMMAND) ROUTINES FOR PLTCCLASS

MAP RADIANCE/DENSITY/CLASS (PHASE 0)
PLOT UNIT HEADING (UTILITY)
REGISTER DETECTION PIXELS (UTILITY)

OUTPUT CLASSIFIED ERTS MAPS ON FILM RECORDER

MAP DAM.FLCCLASS TO TPFS AND BXQT.1
COLLECTOR SYMBOLS FOR REAL ABSOLUTE

CALL PHASE 0 (COMMAND) ROUTINES FOR FLTCCLASS

MAP RADIANCE/DENSITY/CLASS (PHASE 0)
GENERATE & PLOT MARGINAL TICKS (UTILITY)

PLOT UNIT HEADING (UTILITY)
REGISTER DETECTION PIXELS (UTILITY)
MAIN PROGRAMS/ROUTINES

**APRT.S**  DAM. STATUS       DETERMINE STATUS OF DAM PACKAGE RUNS
**APRT.SC** DAM. STATUS-HIA          . HIERARCHY
**APRT.SC** DAM. STATUS/VIRTUAL    . MAP DAM. STATUS TO TPFS & IXOT.I
**APRT.SC** DAM. STATUS-MAP        . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
**APRT.SC** DAM. STATUS-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE

**SPRT.S**  DAM. DITCOP         DISK TO TAPE COPY PROGRAM
**SPRT.SC** DAM. DITCOP/VIRTUAL  . MAP DAM. DITCOP TO TPFS AND IXOT.I
**SPRT.SC** DAM. DITCOP-MAP      . COLLECTOR SYMBOLICS FOR REAL ABSOLUTE
**SPRT.SC** DAM. DITCOP-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
**SPRT.SC** DAM. DITCOP-MAP/VIRTUAL . COLLECTOR SYMBOLICS FOR VIRTUAL ABSOLUTE
**SPRT.SC** DAM. DITO00         CALL PHASE 0 (COMMAND) ROUTINES FOR DITCOP
**SPRT.SC** DAM. DITEXI         TERMINATION ROUTINE
**SPRT.SC** DAM. DITDUP         DUPLICATE DETECTION FILE ONTO TAPE FROM DISK
**SPRT.SC** DAM. DITIVE         VERIFY DETECTION FILE ON TAPE
**SPRT.SC** DAM. DITXQT         INITIALIZATION ROUTINE
MAIN PROGRAMS/ROUTINES

• SETUPS:
  FREE TPFS. (TOO SMALL)  . : SETUPS
  ASQ.T TPFS.F/O/TRK/256 (BIGGER) : SETUPS
  COPY.A DAM..TPFS. ; : SETUPS
  XOR ERSPRTCN . : SETUPS
  D..SETTY C.010
  D..SETTY W.132

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

• SETUPS:
  ADD.E DAM..NEW-DAM . : SETUPS
SETUP HIERARCHY

SETUP

: : :
ERSPRTCN
: : 
NEW-DAM
PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

#SETUPS: FREE TPFS. (TOO SMALL)
#SETUPS: ASQ. T TPFS..F/D/FRK/256 (BIGGER)
#SETUPS: COPY..A DUM..TPFS.
#SETUPS: XOR ERSPTCN
D..S\$TTY C.010
D..S\$TTY W.132

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

#SETUPS: EOF
OAH PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

ERSPRTCN
001

PROGRAM ERSPTCN & OUTPUT PRINT IMAGES / PERFORM PRINT CONTROL DIRECTIVES

HISTORY
--------
E H SCHLOSSER LEC 08/27/75 ORIGINAL CODE
E H SCHLOSSER LEC 12/10/79 ELIMINATE THIRD-WORD J-DESIGNATORS

METHOD
-------
THIS ASSEMBLER PROGRAM MAKES THE PRINT FACILITIES OF EXEC-0 ER PRINTS
AND THE PRINT CONTROL FACILITIES OF EXEC-0 ER PRTCNS DIRECTLY
AVAILABLE FROM THE RUNSTREAM. THIS IS PARTICULARLY USEFUL FOR PRINTING
IMAGES FROM AN 3ADD-ED ELEMENT AND/OR SUBMITTING DEMAND SYMBIONT
(88) CONTROL STATEMENTS FROM AN 3ADD-ED ELEMENT.

PROCESSING OF INPUT TO ERSPTCN IS DETERMINED BY COLUMN 1:

COLUMN 1 PROCESSING
0 SUBMIT TO ER PRTCNS (DEMAND RUN ONLY)
A.I.L.H.M.R.W SUBMIT TO ER PRTCNS (DEMAND & BATCH RUNS)
(BLANK),.*,0.1 PRINT VIA ER PRINTS (1ST CHR IS FORTRAN CARRIAGE CONTROL
(OTHER) GENERATES DIAGNOSTIC FROM SYMBIONT

MACHINE-DEPENDENT CODE
------------------------
WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE
EXEC-0 OPERATING SYSTEM USING 6-BIT FIELD DATA CHARACTERS.
IMPLEMENTING CODE MUST BE RERITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES
---------------------
ER PRINTS & PRINT FIELD DATA IMAGE
ER PRTCNS & PERFORM FIELD DATA PRINT CONTROL DIRECTIVE
ER EXITS & 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:

L-10
MAIN PROGRAMS/ROUTINES

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 '88888D'. (ONLY REMAINS 'O' IF BATCH)

PROCEDURE

$01 . I-BANK
ERSPRCHN TNE .U A4.4
SZ DBATCH . DEMAND: REPLACE 'O' WITH ZERO

READ LA A0 .(COF.BUF)
ER READS . READ CARD IMAGE
SA.H2 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 .'8888' . SKIP NI IF NOT SPACE
J ADV1 . "" == ADVANCE 1 LINE BEFORE PRINTING
TNE .U A1 .'8888' . SKIP NI IF NOT '0'
J ADV0 . "" == ADVANCE 0 LINES BEFORE PRINTING
TNE .U A1 .'8888' . SKIP NI IF NOT '1'
J ADV2 . '0' == ADVANCE 2 LINES BEFORE PRINTING
TNE .U A1 .'8888' . 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 . SUBMIT CONTROL IMAGE
ER PRTCHNS .
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
<table>
<thead>
<tr>
<th>PRINT</th>
<th>SA.S2</th>
<th>A1.PKT</th>
<th>RIGHTMOST 6 BITS OF # LINES TO ADVANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSL</td>
<td>SA.S1</td>
<td>A1.PKT</td>
<td>PREPARE TO STORE NEXT 6 BITS</td>
</tr>
<tr>
<td>SA.S3</td>
<td>SA.S3</td>
<td>A0.PKT</td>
<td>LEFTMOST 6 BITS OF # LINES TO ADVANCE</td>
</tr>
<tr>
<td>LA.U</td>
<td>LA.U</td>
<td>A1.&quot;00000&quot;</td>
<td>LENGTH OF IMAGE IN WORDS</td>
</tr>
<tr>
<td>LA.S1</td>
<td>LA.S1</td>
<td>A1.BUF+0</td>
<td>BLANK OUT ...</td>
</tr>
<tr>
<td>ER</td>
<td>ER</td>
<td>PRINTS</td>
<td>... 1ST CHARACTER</td>
</tr>
<tr>
<td>J</td>
<td>J</td>
<td>READ</td>
<td>... PRINT</td>
</tr>
<tr>
<td>EOF</td>
<td>EOF</td>
<td>EXITS</td>
<td>... TERMINATE PROGRAM</td>
</tr>
<tr>
<td>END</td>
<td>END</td>
<td>ERSPRCH</td>
<td></td>
</tr>
</tbody>
</table>
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAN. ERSPRICH
LIB DAN.
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTES

IN DAM.ERSPRICN
LIB DAM.
PROCESSOR DATA/CHECKOUT & DATA/CHECKOUT MODE: INTERCEPT/CHECK RUNSTREAM

HISTORY
E N SCHLOSSER LEC 12/09/77 REQUIREMENTS

METHOD
This processor allows the user to verify the syntax of both EXEC commands and program commands for a complete DAM package run without actually mounting any tapes or processing any LANDSAT MSS data.

This processor intercepts and reads all card image input in the runstream. It checks all images for valid EXEC commands and valid syntax, and flags errors.

When this processor encounters an 'EXGT' EXEC command for any program in the DAM package, it terminates and initiates that program with the requested options plus a 'D' option (DATA mode) and a 'C' option (CHECKOUT mode).

That program then reads card image input from the runstream. It checks all images for valid commands and valid syntax, and flags errors. It does not assign work files or generate output (other than diagnostics). When the program terminates, it re-initiates the DATA/CHECKOUT processor.

DATA/CHECKOUT mode continues, alternating between this processor and programs in the DAM package until the following card image is read:

RESTRICTIONS

1. The absolute element for this processor must be stored in the DAM program file in real (not virtual) form.

2. The following control card must appear in the run before any reference to this processor:
   SADD DAM.SETUP

3. When in DATA/CHECKOUT mode, syntax errors on an SADD EXEC command abort the run.

4. DATA/CHECKOUT mode may not be initiated or terminated from within an executing program.

L-15
C DATA/CHECKOUT
C END
C
C EXTERNAL REFERENCES
C
C ERPRNT
C
C GLOBAL DECLARATIONS
C
C NONE.
C
C LOCAL DECLARATIONS
C
C TO BE DETERMINED
C
C
C PROCEDURE
C
C
C CALL ERPRNT(1,5,'DATA/CHECKOUT NOT IMPLEMENTED')
STOP
END
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IM DAM.DAT/CHECKOUT
LIB DAM.
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM DATA/CHECKOUT
LIB DAM.
DATA-CHECK

MAIN PROGRAMS/ROUTINES

DATA: DATA/CHECKOUT MODE, DATA/CHECKOUT MODE, DATA/CHECKOUT MODE

L-19
PROCESSOR IOFILE & PSEUDO EXEC COMMAND -- IDENTIFY TAPE/DISK FILE

HISTORY

E M SCHLOSSER \t LEC \t 05/11/79 \t ORIGINAL CODE
E M SCHLOSSER \t LEMSCO \t 07/22/80 \t DUMP NON-CHAR INFO IN OCTAL/BINARY

USAGE

1. THE ABSOLUTE ELEMENT FOR THIS PROCESSOR MUST BE STORED IN THE DAM PROGRAM FILE IN REAL (NOT VIRTUAL) FORM.

2. THE FOLLOWING CONTROL CARDS MUST APPEAR IN THE RUN BEFORE ANY REFERENCE TO IOFILE:
   - USE DAM...(NAME OF DAM PROGRAM FILE)
   - BPRRT.T
   - BASQ.A DAM.
   - ADD DAM.SETUP

SYNTAX

$IDFILE <INTERNAL FILE NAME>

METHOD

GET FILE NAME FROM EXEC-B INFOF BUFFER, GET FACILITY INFO FROM EXEC-B FITENS PACKET, PRINT IN CHARACTER, OCTAL, AND BINARY.

MACHINE-DEPENDENT CODE

ASSUMES 6 CHARACTERS PER INTEGER, ASSUMES 6 BITS PER CHARACTER.
MANIPULATES UNIVAC EXEC-B I/O AND FACILITY PACKETS.
USES UNIVAC FORTRAN V FIELD FUNCTION.

EXTERNAL REFERENCES

- SYSOET \t GET RECORD FROM SYSIN RUNSTREAM
- ERPRINT \t WRITE CHARACTER BUFFER TO PRIMARY OUTPUT DEVICE
- ERFILM \t RETRIEVE FACILITY ASSIGNMENT INFORMATION PACKET
- CBINIT \t INITIALIZE CHARACTER BUFFER
- GETCHR \t GET CHARACTER FROM CHARACTER STRING
- GETICE \t GET INTEGER-CHARACTER-EQUIVALENT FROM CHAR STRING
- CB4CST \t CHARACTER BUFFER FOR CHARACTER STRING

L-20
C

EXCEPTIONS


1. THE FOLLOWING CONDITIONS GENERATE DIAGNOSTICS:
   Module not invoked as processor in TPFS.
   File name not specified.
   File not assigned.

GLOBAL DECLARATIONS


INCLUDE NULCST.LIST  # DEFINE NULL CHARACTER STRING
INCLUDE ICBUFI.LIST   # DECLARE CHARACTER BUFFER # 1
INCLUDE ICBUF2.LIST   # DECLARE CHARACTER BUFFER # 2
INCLUDE ICBUF3.LIST   # DECLARE CHARACTER BUFFER # 3
INCLUDE ASMOEF.LIST   # DEFINE ASSEMBLER PARTIAL WORD MNEMONICS

LOCAL DECLARATIONS


INTEGER INFOR(10)  # UNIVAC EXEC-8 INFORS BUFFER
INTEGER FPKT(13)   # UNIVAC EXEC-8 FACILITY INFO PACKET
INTEGER INSTAT     # INPUT STATUS FROM SYSOET
INTEGER LENGTH     # INPUT LENGTH IN CHAR FROM SYSOET
INTEGER NW         # WORD NUMBER, STARTING FROM 0 AT LEFT
INTEGER NC         # CHARACTER NUMBER, STARTING FROM 1 AT LEFT
INTEGER NB         # BIT NUMBER, STARTING FROM 0 AT LEFT
INTEGER KHAR       # CHARACTER
INTEGER NICE       # INTEGER-CHARACTER-EQUIVALENT
INTEGER IBIT       # BIT

PROCEDURE


GET UNIVAC EXEC-8 INFORS BUFFER CONTAINING EXEC COMMAND SPEC(S)
   CALL SYSOET(INSTAT, INFOR, LENGTH)

CHECK IF INVOKED AS A PROCESSOR IN TPFS.
   IF(INSTAT.EQ. ' ') AND (INFOR(3), EQ. 'IDFILE') GO TO 150
   CALL ERPRNT(1,3, 'PROGRAM NOT FOUND')
   GO TO 900

150 CONTINUE

CHECK IF FILE NAME WAS SPECIFIED
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(ASM58(INFOR(4)).NE.1) IFPKT(2)=INFOR(6)
C C
C GET/CHECK EXEC-Q FACILITY INFO IN PACKET
C CALL ERFP3NT(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+1')
CALL CB4IN (ICBUF1, NW, 'O')
CALL CBINIT(ICBUF2)
CALL CB4CST(ICBUF2, ICBUF1)
CALL CB4CST(ICBUF3)
CALL CB4CST(ICBUF3, ICBUF1)
CALL CB4CST(ICBUF3, ' ', (1),(2))
CALL CB4CST(ICBUF2, ' ', (1),(2))
CALL CB4CST(ICBUF3, ' ', (1),(2))
DO 430 NC=1,8
CALL GETCHR(KCHAR, IFPKT(NW+1),(NC))
CALL CB4CST(ICBUF1, KCHAR, (1),(1))(1)
CALL CB4CST(ICBUF1, ' ', (1),(6))
CALL GETICE(NICE, IFPKT(NW+1),(NC))
CALL CB4IN(ICBUF2, NICE/6.1)
CALL CB4IN(ICBUF2, ' ', NICE/6.1)
CALL CB4CST(ICBUF2, ' ', (1),(5))
DO 410 NB=0.9
IBIT=FLD(NB,1,KCHAR)
CALL CB4IN(ICBUF3, IBIT,1)

L-22
CONTINUE
CALL CONCST1ICBUF3.

430 CONTINUE
CALL ERPRNT(2, 10, ICBUF1)
CALL ERPRNT(1, 10, ICBUF2)
CALL ERPRNT(1, 10, ICBUF3)

460 CONTINUE

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 TERMINATE
C
900 STOP
END
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM.IDFILE
LIB DAM.
MAIN PROGRAMS/ROUTINES

IN DAM, IODEFILE
LIB DAM.
PROCESSOR LOCATE A PSEUDO EXEC COMMAND: LOCATE FILE ON MULTIFILE BIP TP

HISTORY

E H SCHLOSSER LEC 12/22/70 ORIGINAL CODE
E H SCHLOSSER LEC 04/20/70 CHECK '-' IN SCENE NUMBER
E H SCHLOSSER LEC 12/17/70 SUPPORT BB OR BST BUFFER FORMAT
E H SCHLOSSER LEMSCO 07/22/80 REPLACE KCHAR WITH GETCHR
E H SCHLOSSER LEMSCO 06/27/80 CHK BUF FMT FROM FLINFO

METHOD

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

RESTRICTIONS

1. THE ABSOLUTE ELEMENT FOR THIS PROCESSOR MUST BE STORED IN THE DAM PROGRAM FILE IN REAL (NOT VIRTUAL) FORM.

2. THE FOLLOWING CONTROL CARDS MUST APPEAR IN THE RUN BEFORE ANY REFERENCE TO LOCATE:
   SADD DAM.SETUP
   SASO.<OPTIONS> 3..US.<REEL NUMBER>

SYNTAX

LOCATE!E1 3..<LANDSAT STRIP NUMBER>

MACHINE-DEPENDENT CODE

VERRYY!

EXTERNAL REFERENCES

ERPCY 8 GET PART OF EXEC-8 PROGRAM CONTROL TABLE
EREXIT 8 TERMINATE PROGRAM
ERFACL 8 RETRIEVE FACILITIES ASSIGNMENT INFORMATION
ERPRINT 8 PRINT IMAGE ON TTY OR LINE PRINTER (FILEDATA)
ERRR 8 ERRS TERMINATE
ERREAD 8 READ IMAGE FROM TTY OR CARD READER (FILEDATA)
GLOBAL DECLARATIONS

INTEGER JPCT(25)  8 EXEC-0 PROGRAM CONTROL TABLE (1ST 25 HDS)
INCLUDE XGDEF.LIST  8 DEFINE STATUS OF GBT OPTIONS
INCLUDE ASHDEF.LIST  8 DEFINE UNIVAC ASSEMBLER PARTIAL HDS IN FORTRAN V
INCLUDE KOMIO.LIST  8 FORTRAN MANIPULATION OF ASSEMBLER I/O PACKETS
INCLUDE FIDDEF.LIST  8 RHEMONICS FOR LOCATIONS IN FIDDEF-FORMAT BUFFER

LOCAL DECLARATIONS

INTEGER IPAGE(14)  8 EXEC-0 PROCESSOR INFOR BUFFER
INTEGER IDFIL3(10)  8 FILE INFORMATION IN FIDDEF FORMAT FOR FILE 3
INTEGER HPKT3(9)'/3', 'O' '/ 8 EXEC-0 TAPE I/O PACKET FOR FILE 3'
INTEGER IOBUF(40)  8 ID RECORD BUFFER
DATA IOBUF(11/0/ 8 ELIMINATE SPURIOUS COMPILER DIAGNOSTICS
DATA IANBUF(80)  8 ANNOTATION RECORD BUFFER
DATA IANBUF(11/0/ 8 ELIMINATE SPURIOUS COMPILER DIAGNOSTICS
INTEGER READ' '/ 'SK' '/ READ
INTEGER REND'/ 'SL' '/ READ BACKWARD
INTEGER REND'/ 'SL' '/ RENIND

PROCEDURE

RETRIEVE PROGRAM CONTROL TABLE & CONTROL CARD SPECIFICATIONS

CALL .RPCT(25,JPCT)
CALL INFOGUID

CHECK FOR 'E' OPTION

IF
1 (KOPT1('ERROR'),EQ.1).AND.
6 (ASHMJPCT(11/1).NE.21) .1 .PREVIOUS EXECUTION DID NOT ERROR TERMINATE
6 CALL EREXIT

CHECK IF TAPE 1 IS ASSIGNED TO 3.
CALL FLINFO(IDFIL3, '2', '00')
IF(IDFIL3(FIDCQT).EQ.'NULL') GO TO 810  8 NOT ASSIGNED
IF(IDFIL3(FIDCQT).NE.'TAPE') GO TO 820  8 NOT TAPE

C
C READ NEXT TAPE BLOCK AND CHECK IF IT IS ID RECORD
C
CALL IDENT3(8300)
IF(NCCT.LE.NSTRIp) GO TO 800

C
C REWIND TAPE, THEN READ & CHECK FIRST ID RECORD
C
300 CALL 103(REM,8320)
320 CALL IDENT3(8340)
GO TO 800
340 CALL 103(REM,8360)
360 CALL IDENT3(8800)

C
C CHECK IF ID RECORD IS FOR SPECIFIED STRIP
C
500 IF(NCCT.EQ.NSTRIp) GO TO 800
IF(NCCT.GT.NSTRIp) GO TO 850

C
C FIND NEXT END-OF-FILE MARK
C
CALL 103(R,8800)
600 CALL 103(R,8700)
GO TO 800

C
C READ BLOCK AFTER EOF & CHECK IF IT IS ID RECORD
C
700 CALL IDENT3(8830)
GO TO 800

C
C FLAG TAPE ERROR(S)
C
800 CALL ERPRINT(1,2,'TAPE ERROR')
GO TO 890
810 CALL ERPRINT(1,3,'FILE NOT ASSIGNED')
GO TO 890
820 CALL ERPRINT(1,3,'FILE NOT TAPE')
GO TO 890
830 CALL ERPRINT(1,3,'STRIP NOT ON TAPE')
890 IF(ASISHSTJ.PCT(29).EQ.4) GO TO 890  8 ONLY IN DEMAND
IF(NCCTOTP('CONTINUE').EQ.1) GO TO 493
CALL ERRR  8 ERRS TERMINATION!

C
C POSITION TAPE BEFORE ID RECORD
C
990 CALL 103(RE,8800)
SUBROUTINE INFOR(S)

CALL ERREAD(S990,IMAGE,MSG)
IFn(IMAGE(1).NE.'LOCATE') Go TO 890
IFn(IXSTOPT('CONTIN').EQ.1) 8 C OPTION SPECIFIED ...
& NPKT(11)=IMAGE(1)
IFn(IMAGE(1).NE.NPKT(11)) Go TO 890

FLD(16,30,IMAGE(7)).NE.'0') Go TO 870
MCSTRIP=FLD(8,IMAGE(7))=2888883'
IFn(INSTRP.LT.1).OR.INSTRP.GT.41) Go TO 870
RETURN

SUBROUTINE IDENT3(S)

READ (HOPEFULLY) ID RECORD

1DBUF(10)=0
1OSIZE(NPKT3)=40
1OADDR(NPKT3)=LOC(1DBUF)
CALL 103'R'.8000'
If(IONWDS(NPKT3).GT.14) Go TO 800 8 TOO LONG TO BE AN ID RECORD
If(IDFIL3.1F1DBFH).NE.'BST') Go TO 300
DO 200 NBY=1,38
  CALL GETBYT(1BYT,1DBUF.NBY)  
  If(1BYT.GT.255) IDFIL3.1F1DBFH='88'
  200 CONTINUE
If(IDFIL3.1F1DBFH).NE.'BST') CALL ERPRINT:
  6 1.S.**'BST BUF FMT FROM FLINFO WRONG**'
300 IF(IDFIL3.1F1DBFH).EQ.'8W') CALL BST488(1DBUF,1DBUF.80)
CALL CST488(1DBUF,1DBUF.80)
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

READ ANNOTATION RECORD

LOADR(NPKT3)=LOC(IANB.)
CALL 1031R(NPKT3)
IF(IOFIL3(IF0BFM).EQ."NO") CALL BST488(IANBUF.1ANBUF.300)
CALL CST488(IANBUF.1ANBUF.300)
IF(IOTOPT("LIST").EQ.0) GO TO 600 8 L OPTION MEANS LIST TAPE HEADERS
CALL ERPRINT(1.7.IANBUF(1))
CALL ERPRINT(1.9.IANBUF(1))
CALL ERPRINT(1.12.IANBUF(1))

GET/CHECK STRIP NUMBER

800 CALL GETCHR(KHTEMP.1DBUF.5)
IF(KHTEMP.NE."-" ) CALL GETCHR(KHTEMP.1DBUF.6)
IF(KHTEMP.NE."-" ) GO TO 800
CALL GETICE(NICE.1DBUF.14)
NCCT=NICE-ICE("0")
IF((NCCT.LT.1).OR.(NCCT.GT.4)) GO TO 800
CALL GETCHR(KHTEMP.1DBUF.15)
IF(KHTEMP.NE."-" ) GO TO 800
CALL GETICE(NICE.1DBUF.18)
NCCT=NCICE("0")
IF(NCCT.NE.4) GO TO 800
RETURN 3 NORMAL RETURN
800 RETURN 1 3 ERROR RETURN

SUBROUTINE 1031NXTFUN,S

IF(NXTFUN.NE.LSTFUN) CALL ER10(NPKT3)
IF(IOFUNC(NPKT3).EQ."NO") CALL ER10(NPKT3)
LSTFUN=NXTFUN
CALL ER10(NPKT3)
IF(IOFUNC(NPKT3).EQ."NO") CALL ER10(NPKT3)
RETURN 2 8 EOF ENCOUNTERED

END
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM.Locate
LIB DAM.
QAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN QAM.LOCATE L19 DAM.

LOCATE-HAP/VIRTUAL 001
PROGRAM USHAP 3 SWAP TAPE DRIVE UNITS

HISTORY
--------
E H. SCHLOSSER  LEC  12/09/77  REQUIREMENTS

METHOD
-------
THIS PROCESSOR ALLOWS THE USER TO SWAP TAPE DRIVE UNITS. USHAP
GENERATES ALL THE NECESSARY EXEC COMMANDS, WRITES THEM TO TEMPORARY
FILE 20, AND THEN ADDS FILE 20 TO THE RUNSTREAM.

RESTRICTIONS
------------
1. THE ABSOLUTE ELEMENT FOR THIS PROCESSOR MUST BE STORED IN THE DAM PROGRAM
FILE IN REAL (NOT VIRTUAL) FORM.
2. THE FOLLOWING CONTROL CARD MUST APPEAR IN THE RUN BEFORE ANY REFERENCE
TO THIS PROCESSOR:
   $ADD DAM SETUP
3. A TAPE REEL MAY NOT BE SWAPPED MORE THAN ONCE IN ANY RUN.
4. IF TWO TAPE FILES ARE SPECIFIED, THEY MUST HAVE THE SAME DENSITY
AND NUMBER OF TRACKS.

SYNTAX
------
\$USHAP3.E1 (<TAPE FILE NAME>1, <TAPE FILE NAME>1)

EXTERNAL REFERENCES
---------------------
ERPRINT

GLOBAL DECLARATIONS
---------------------
NONE.

LOCAL DECLARATIONS
--------------------
(TO BE DETERMINED)
CALL ERPRNT(1,4,"USHAP NOT IMPLEMENTED")
STOP
END
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM.USMAP
LIB DAM.
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAN.USMAP
LIB DAN.
0000 DAM ERTS-DUP(7703)
0001 $JUMP BEGIN
0002 $USAGE: THIS EXEC B COMMAND STREAM COPIES ERTS TAPE INN TO OUT, IGNORING
0003 $USAGE: FRAME COUNT ERRORS. IF UNRECOVERABLE PARITY ERRORS OCCUR IT
0004 $USAGE: REWINDS BOTH TAPE AND TRIES AGAIN. IF AND WHEN AN APPARENTLY
0005 $USAGE: SUCCESSFUL COPY IS MADE, THE COPY IS IDENTIFIED AND VERIFIED, AND
0006 $USAGE: THE RUN TERMINATED. (E M SCHLOSSER)
0007 $USAGE: 
0008 $USAGE: SEE DAM.EXP-ERTS-DUP FOR INSTRUCTIONS
0009 $USAGE: SEE DAM.RUN-ERTS-DUP FOR SAMPLE RUNSTREAM
0010 $USAGE: 
0011 $BEGIN:
0012 $SETC1. DON'T ABORT RUN ON ERRORS TERMINATION
0013 $USE 3.INN.
0014 $REWWIND 3.
0015 $XOT ERTSIOC
0016 $REWWIND INN.
0017 $REWWIND OUT.
0018 $LOG DAN ERTS-OUP(7703) -- FIRST TRY
0019 $COPY.MN INN..OUT..S . MUST HAVE 2342 BLOCKS PER DATA FILE ***** FIRST TRY *****
0020 $TEST TO/1/52
0021 $JUMP OK
0022 $REWWIND 3.
0023 $XOT.E ERTSIOC
0024 $REWWIND INN.
0025 $REWWIND OUT.
0026 $LOG DAM ERTS-DUP(7703) -- SECOND TRY
0027 $COPY.MN INN..OUT..S . MUST HAVE 2342 BLOCKS PER DATA FILE ***** SECOND TRY *****
0028 $TEST TO/1/52
0029 $JUMP OK
0030 $SWAP:
0031 $TEST TE/0/TE
0032 $JUMP END . (ONLY SWAP ONCE)
0033 $USE OLDINN.INN.
0034 $USE OLDOUT.OUT.
0035 $USE INN.NEWINN.
0036 $USE OUT.NEWOUT.
0037 $JUMP END
0038 $OK:
0039 $FREE INN.
0040 $CLOSE OUT.
0041 $USE 3.OUT.
0042 $XOT PICTAB
0043 EXIT
0044 $FREE OUT.
0045 $FIN
0046 $END:
0047 $SET.A 1/TE . RESET & FLAG
PROGRAM ERTSIOC  & PRINT ID/HEADER AND ANNOTATION RECORD INFO

HISTORY
-------

E H SCHLOSSER   LEC   05/29/73   ALGORITHM CODING
J C CRISP      LEC   10/24/78   REVISE TO USE CHAR BUFFER Routines

METHOD
-----

CALL OPEN5 TO READ ID/HEADER AND ANNOTATION RECORDS. CALL ROUTINES
TO PRINT INFORMATION EXTRACTED AND COMPUTED. BUILD BUFFER TO
PUT ERTS ID ON TAIL SHEET.

ERTS CONVENTIONS FOR ATTITUDE AND HEADING:

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

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

UTILIZES UNIVAC EXEC 8 ER CSFS AND ER PRINTS

EXTERNAL REFERENCES
---------------------

PSTART   8 INITIALIZE PROGRAM
OPEN5    8 OPEN INPUT MBS/REO FILE ASSIGNED TO UNIT 3
IDOL5    8 PRINT SHORT ID FOR LOGICAL UNIT 3
IDERT5   8 PRINT COMPLETE ERTS SCENE ID
CBINIT   8 INITIALIZE CHARACTER BUFFER
CBWCL5   8 CHARACTER BUFFER FOR CHARACTER STRING
CBVCL5   8 CHARACTER BUFFER FOR INTEGER
ERCSF   8 SUBMIT EXEC COMMANDS
PSTOP    8 PROGRAM TERMINATION

EXCEPTIONS
----------

NONE

GLOBAL DECLARATIONS
---------------------

INCLUDE KONXOT.LIST  8 COMMON PROGRAM SWITCHES.COUNTERS
INCLUDE KONNER.LIST
INCLUDE ICBUFI.LIST

LOCAL DECLARATIONS

NONE

PROCEDURE

INITIALIZE PROGRAM AND OPEN INPUT MSS FILE ON UNIT 3

CALL PSTART("DAN ERTSIOC (0009)"
CALL OPEN 3
IF (NDATAC.NE.0) GO TO 900

IDENTIFY UNIT 3 HARDWARE/SENSOR/SCENE/DATE/ETC

CALL EPRNT (1)"
CALL IDLU3 (6)
CALL IDERTS (6)

PUT SCENE ID INTO UNIVAC EXEC-B SYSTEM LOG SO IT GETS PRINTED ON TAIL SHEET

CALL CBINIT (ICBUFI)
CALL CB4CST (ICBUFI, "BLOG", (1), (9))
CALL CB4CST (ICBUFI, NERSAT, (1), (12))
CALL CB4CST (ICBUFI, "SCENE", (1), (7))
CALL CB4IN (ICBUFI, NERTS(1), 1)
CALL CB4IN (ICBUFI, NERTS(2), 9, 0)
CALL CB4CST (ICBUFI, ";")
CALL CB4IN (ICBUFI, NERTS(3), 5, 0)
CALL CB4CST (ICBUFI, "CCT", (1), (7))
CALL CB4IN (ICBUFI, NCCT, 11)
CALL CB4CST (ICBUFI, "OF", (1), (4))
CALL CB4IN (ICBUFI, NCCTOT, 11)
CALL CB4CST (ICBUFI, ";", (1), (3))
CALL ERSF (NAD, ICBUFI)

TERMINATE

900 CALL PSTOP("***PLEASE FREE 3. ORREWIND 3. ORLOCATE 3.")
END
ERTSIDC HIERARCHY

ERTSIDC

PSTART OPEN3 PSTOP
PROGRAM ERTS10C/VIRTUAL

HISTORY

E N SCHLOSSER  LEC  08/02/74  ORIGINAL CODE
E N SCHLOSSER  LEC  11/08/79  SNAP.P2(N): NO 'N' IN DEMAND

METHOD

CONSTRUCT SNAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT SNAP COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE SNAP & B:CT COMMANDS TO TEMPORARY FILE 28.
ADD TEMPORARY FILE 28, TO RUNSTREAM.

MACHINE-DEPENDENT CODE

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

EXTERNAL REFERENCES

ER CSFS   8 FUNCTION TO SUBMIT EXEC-D CONTROL STATEMENT
ER IOVS   8 INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS  8 TERMINATE PROGRAM EXECUTION
DAN-ERTS10C-MAP 8 SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAN-SYS-MAPOPT 8 STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAN. IS SASD-O & SREP-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

  B:CT OPTIONS ARE PRE-LOADED BY EXEC INTO REGISTER AS IN
MASTER BIT NOTATION.

LOCAL DECLARATIONS

AXRS

S1001 . O-BANK
SSSH FORM 6.6.6.16

ERTSIDC+ MAP.11.1 FORM 6.6.6.16 LABEL 1 HD. FORTRAN. FIELDATA
LADINO "SOFT" LADINO "SOFT"
HAP=OFDF SSHN 000.0.0.0 DATA. 9 MBS. . FIELDATA
HAPINO "8XQTS: HAP.FZN DNH.ERTSIDC-MAP.ERTSIDC . 18XQTS"
ADDSENS SSHN 000.0.0.0 ADDINO "8XQTS: ADD . DNH.SYS-HAPOPT . 18XQTS"
HXTSDF SSHN 000.0.0.0 HXTINO "8XQTS: HXT . 18XQTS"

EOFSDF 0 . END-OF-FILE STOP WORD
PF FORM 12.6.16
CSFADO "5ACQ.7 20." .
CSFADO "BADD 20." .
SAVREO RES 1

10POT 1600 "20", MS 33.LABSDF.‘0’ 0

PROCEDURE

- - - - - - - - - -

S1011 . I-BANK
ERTSIDC+ ERTSIDC+ LAU . A0. . . . AB = . . .
TNEU A4.4 . SKIP NEXT INST IF A4<24 (NOT DEMAND)
SA.SZ A0. HAPINO+2 . DEMAND! BLANK OUT N OPTION
LA A0. (CSFADO) . ADDRESS OF BASO IMAGE
ER CSF6 . DO IT
SA A0. SAVREO . STORE 6
PRINT (PF 0.1.SAVREO) . PRINT BASO STATUS

GETOPT . LOAD OPT LETTERS INTO A2.A3.A4

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

WRITE LA A0. (10POT) . ADDRESS OF I/O PACKET
ER I0NS . WRITE SDF IMAGES TO 20.

ADD LA A0. (CSFADO) . ADDRESS OF BADD IMAGE
ER CSF6 . DO IT
ER EXITS

END ERTSIDC
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM.ERTSIDC/...HTags/DAM...SYS-BLOCK
LIB DAM.
DAN PACKAGE APPENDIX L
RAIN PROGRAMS/ROUTINES

IN DAN.ERFSIOC/VIRTUAL
PROGRAM PICTAB & PICTURE/TABULATE/DISPLAY/LIST/Factor MSS DATA

HISTORY

E H EKnOSER LEC 07/02/73 ORIGINAL CODE
J C CRISP LEMSCO 05/28/80 UPGRADE DOCUMENTATION & ADD KONKS

METHOD

This program pictures, displays, lists, tabulates, correlates, factors, models, and/or partitions raw or transformed data from Landsat BIP('X') or BIL('AM' OR 'PM') computer-compatible tape assigned to logical unit 3.

Origins & windows are specified by scanner, geographic, or uth coordinates, in order to minimize tape accesses. The first origin should have the lowest line numbers, and each succeeding origin progressively higher scan line numbers.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

NVIATO & NAME 'VIA' 'TO' ROUTINES
VIATO & CALL 'VIA' 'TO' ROUTINES
VIA TO EXTERNAL PICDOO, PICXQT

EXCEPTIONS

1. If control has not been executed in the current run prior to PICTAB and satisfactorily adjusted a control network for the scene to be processed by PICTAB, then PICTAB will use nominal registration parameters and nominal scene center.

2. If PICXQT does not call NVIATO to change the 'VIA' and/or 'TO' routines, then PICTAB will call to PICXQT in an endless loop!

GLOBAL DECLARATIONS

ORIGINAL PAGE IS OF POOR QUALITY
INCLUDE KONKGFT.LIST  8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONLOGG.LIST  8 COMMON LOG FILE BUFFER, I/O PKT, POINTERS
INCLUDE KONLUGJ.LIST  8 COMMON POINTERS/FLAGS FOR UNIT 3
INCLUDE KONLU3.LIST  8 COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
INCLUDE KONLWU.LIST  8 COMMON INPUT WINDOW PACKETS
INCLUDE KONLW3.LIST  8 COMMON OUTPUT WINDOW PACKETS
INCLUDE KONNNER.LIST  8 COMMON ERTS SCENE PARAMETERS
INCLUDE KONKLS.LIST  8 COMMON CLASSIFICATION INFO
INCLUDE KONKFIT.LIST  8 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KONKIRT.LIST  8 COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
INCLUDE KONHALT.LIST  8 COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS
INCLUDE KONSYM.LIST  8 COMMON SYMBOL TABLE
INCLUDE KONKS.LIST  8 COMMON COLOR SCREEN PARAMETERS
INCLUDE KONKBL.LIST  8 COMMON MULTI-PURPOSE TABLE

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

CALL NVIATO( PIC000,PICXQT)  8 FIRST CALL IS VIA PIC000 TO PICXQT
100 CONTINUE
    CALL VIATO
    GO TO 100
END   8 (STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE)
CAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROGRAM PICTAB/VIRTUAL

HISTORY

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

METHOD

CONSTRUCT $MAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT $XOT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE $MAP & $XOT 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-9 OPERATING SYSTEM USING 8-BIT FIELD DATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

ER CSFS 9 FUNCTION TO SUBMIT EXEC-9 CONTROL STATEMENT
ER IOWS 9 INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS 9 TERMINATE PROGRAM EXECUTION
OAM.PICTAB-MAP 9 SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
OAM.SYS-MAPOPT 9 STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS DASO-D & DPREP-D.

GLOBAL DECLARATIONS

1. 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

2. $XOT OPTIONS ARE PRE-LOADED BY EXEC INTO REGISTER A5 IN
(MASTER BIT NOTATION.

LOCAL DECLARATIONS

AXIS

$100 . D-BANK

SSSH  FORM  6,6,8,10

LABSDF  SSSH  050.1. 'F'.0  . LABEL 1 WD. FORTRAN, FIELDATA

LASHMO  '8XQTS'

MAPSDF  SSSH  000.9.0.0  . DATA, 9 WDS, FIELDATA

MAPIMO  '8XQTS: MAP, FZN DAM, PICTAB-MAP, PICTAB  '8XQTS'

ADDSDF  SSSH  000.9.0.0

ADDIMO  '8XQTS: ADD DAM, SYS-MAPOPT  '8XQTS'

XQTSDF  SSSH  000.9.0.0

XQTIMO  '8XQTS: XQT.1  PICTAB  '8XQTS'

EOFSDF  '0  . END-OF-FILE STOP WORD

PF  FORM  12.8.18

CSFASO  'BASO.T 20.

CSFADD  '8ADD 20.

SAPREO  RES  1

IOPKT  1500  '20'.WS 33, LABSDF, '0' 0

PROCEDURE

$101 . I-BANK

PICTAB+  LAU  '100.'  . AO = '100.'

tne.u  49.4  . SKIP NEXT INST IF 44<44 (NOT DEMAND)

SA.S2  40.MAPIO+2  . DEMAND BLANK OUT N OPTION

LA  40.(CSFASO)  . ADDRESS OF BASO IMAGE

ER  CSFDS  . DO IT

SA  40.SAPREO  . STORE &

PRINT  (PF 2.1.SAVREO)  . PRINT BASO STATUS

GETOPT  . LOAD OPT LTRS INTO A2.A3.A4

PUTOPT  DO  XQTIMO+2  . STORE OPTION LETTERS INTO 8XTQ IMAGE

SA  A4.XQTIMO+4  . (3 WORDS -- MAX 10 OPT LETTERS)

WRITE  LA  40.(IOPKT)  . ADDRESS OF I/O PACKET

ER  10MS  . WRITE SDF IMAGES TO 20.

ADD  LA  40.(CSFADD)  . ADDRESS OF 8ADD IMAGE

ER  CSFDS  . DO IT

ER  EXITS

END  PICTAB
PICTAB OVERLAY STRUCTURE

HISTORY

E H SCHLOSSER  LEC  03/19/79  ORIGINAL CODE
E H SCHLOSSER  LEC  07/14/79  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      LEMS CO  01/10/80  IF, FI & OPTIMIZE OVERLAYS
J C CRISP      LEMS CO  05/16/80  CROSSTAB, CHANGE DISTAB TO KMOTAB

LDB DAM.

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

UTILITIES FOR MAIN

SEQ S-FLINFO*.(S-MAIN)
IN DAM.FLINFO . GET FILE DESCRIPTIVE INFORMATION

SEQ S-R3TASCRT9*.(S-MAIN)
IN DAM.R3TREC . READ ONE RECORD FROM TAPE (UNIT 3)
IN DAM.CST4AS . CHARACTER STRING FOR ASCII
IN DAM.ISRTBA . INTEGER BUBBLE SORT ASCENDING

MONITOR FOR PHASE 0.1.2.9 COMMANDS

SEQ S-PICO129*.(S-FLINFO,S-R3TASCRT9)
IN DAM.PICO00 . CALL USER-SPECIFIED PHASE 0 ROUTINE
IN DAM.PICO129 . CALL PREVIOUSLY NAMED PHASE 1/2/9 'TO' ROUTINE
IN DAM.NTABS/DAM . DAM UNIT # TABLE GOES IN SAME SEQ W/ FORTRAN I/O

UTILITIES FOR PHASE 0.1.2.9 COMMANDS

SEQ S-READS*.(S-PICO129)
IN DAM.READS . 'READ' INTO UNIT 5 BUFFER
IN DAM.GETS . 'GET' FREE-FORMAT FIELD FROM UNIT 5 BUFFER
IN DAM.WARN . 'PROCESS WARNING DIAGNOSTIC FOR UNIT 5 FIELD
IN DAM.SPANS . 'ENABLE/DISABLE SPANNING FOR UNIT 5

SEQ S-OPNCRT9*.(S-PICO129)
IN DAM.OPRRI . OPEN ALT PRT FILES
IN DAM.CLOSPR . CLOSE ALT PRT FILES

SEQ S-CALCRP*.(S-PICO129)

L-50
U CAN PACKAGE APPENDIX L

MAIN PROGRAMS/ROUTINES

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

PHASE 0, 1, 2, 9 COMMANDS (FORTRAN I/O ALLOWED)

SEQ S-XQTEXI* . (S-READS, S-OPNCPLR, S-CALCROP)
   IN OAM.PICQXT . PICTAB INITIALIZATION ROUTINE
   IN OAM.PICQXI . PICTAB TERMINATION ROUTINE
SEQ S-PSTART* . (S-XQTEXI)
   IN OAM.PSTART . GENERAL INITIALIZATION ROUTINE
SEQ S-OPEN3* . (S-XQTEXI)
   IN OAM.OPEN3 . OPEN INPUT SCAN DATA FILE (UNIT 3)
SEQ S-OP3DSK* . (S-OPEN3)
   IN OAM.OP3DSK . OPEN INPUT -- DISK IN PXBDEF FMT (UNIT 3)
SEQ S-OP3BIP* . (S-OPEN3)
   IN OAM.OP3BIP . OPEN MSS DATA IN BIP FMT (UNIT 3)
SEQ S-OP3MOP* . (S-OPEN3)
   IN OAM.OP3MOP . OPEN MSS DATA IN MOP FMT (UNIT 3)
SEQ S-03TORMOR* . (S-03MOP)
   IN OAM.03TORM . MOP FMT TAPE DIRECTORY RECORD (UNIT 3)
SEQ S-03ANOT* . (S-OP3MOP)
   IN OAM.03ANOT . MOP FMT ANNOTATION RECORDS (UNIT 3)
SEQ S-03SZAM* . (S-03ANOT)
   IN OAM.03SZAM . SIZE AND INPUT WINDOW FOR AM TAPES
SEQ S-03SZPN* . (S-03ANOT)
   IN OAM.03SZPN . SIZE AND INPUT WINDOW FOR PM TAPES
SEQ S-03SZAR* . (S-03ANOT)
   IN OAM.03SZAR . SIZE AND INPUT WINDOW FOR AR TAPES
SEQ S-03SZPR* . (S-03ANOT)
   IN OAM.03SZPR . SIZE AND INPUT WINDOW FOR PR TAPES
SEQ S-03ANCL* . (S-OP3MOP)
   IN OAM.03ANCL . MOP FMT ANCILLARY RECORDS (UNIT 3)
SEQ S-LOREGB* . (S-XQTEXI)
   IN OAM.LOREGB . LOAD REGISTRATION PARAMETERS FROM UNIT 8
SEQ S-CLOSTOP* . (S-XQTEXI)
   IN OAM.CLOSE3 . CLOSE INPUT SCAN DATA FILE (UNIT 3)
   IN OAM.PSTOP . GENERAL TERMINATION ROUTINE

L-91
MAIN PROGRAMS/ROUTINES

SEO 5-HELP*. (S-READS,S-OPNCLPR,S-CALCROP)
IN DAM.KNDCLE . CLEAR WARNINGS/ERRORS
IN DAM.KNDEXP . EXPLAIN PROGRAM/COMMAND
IN DAM.KNDFI . END IF...FI BLOCK
IN DAM.KNDFF . BEGIN IF...FI BLOCK
IN DAM.KNDNEW . PRINT NEWS
IN DAM.KNDEXX . CONDITIONALLY PERFORM NEXT COMMAND
IN DAM.KNDOFF . TURN OFF NODE SWITCHES
IN DAM.KNDOON . TURN ON NODE SWITCHES
IN DAM.KNPOKE . PECK
IN DAM.KNPREN . RENUMBER (GET/CHECK NEW WINDOW SEQUENCE NUMBER)
IN DAM.KNPTIM . PRINT CLOCK TIME & CHARGE TIME

SEO 5-GEOMETRY*. (S-READS,S-OPNCLPR,S-CALCROP)
IN DAM.KNDAI . ALIGN COORDINATE SYSTEMS
IN DAM.KNOMER . GET/CHECK TRANSVERSE MERCATOR CENT MERIDIAN
IN DAM.KNODIC . GET/CHECK TICK INTERVALS
IN DAM.KNODIN . GET/CHECK WINDOW ENVELOPE/VERTICES
IN DAM.KNODON . GET/CHECK UTM PROJECTION ZONE

SEO 5-SPECS-9*. (S-READS,S-OPNCLPR,S-CALCROP)
IN DAM.KNCH . GET/CHECK RAW/TRANSFORMED SCANNER CHANNEL(S)
IN DAM.KNDMA . GET/CHECK PAGE HEADING(S)
IN DAM.KNOMAM . GET/CHECK TRANSFORMATION/MATERIAL NAME
IN DAM.KNOR . GET/CHECK WINDOW ORIGIN
IN DAM.KNORAD . GET/CHECK RADIANCE LIMITS
IN DAM.KNOSPA . GET/CHECK WINDOW SPACING
IN DAM.PICD9 . DISPLAY RADIANCE/GRADIENT/CLASS (PHASE 9)
IN DAM.PICL9 . LIST RADIANCE/GRADIENT/CLASS (PHASE 9)
IN DAM.PICPA9 . PARTITION FACTOR SPACE (PHASE 9)
IN DAM.PICP9 . PICTURE RADIANCE/GRADIENT/CLASS (PHASE 9)
IN DAM.PICPR9 . PROFILE RADIANCE (PHASE 9)
IN DAM.PICTO9 . TOTAL TABULATIONS (PHASE 9)

SEO 5-MISC*. (S-READS,S-OPNCLPR,S-CALCROP)
IN DAM.KNCL . GET/CHECK COLORS
IN DAM.KNDC . GET/CHECK NUMBER OF OUTPUT COPIES
IN DAM.KNDCRO . CROSSTABULATE COLOR & INTENSITY
IN DAM.DISHIS . HISTOGRAM PREVIOUSLY DISPLAYED DATA (PHASE 9)
IN DAM.KNDINT . GET/CHECK INTENSITY
IN DAM.KNDLIN . GET/CHECK LINEAR TRANSFORMATION WEIGHTS/DAWN/BIAS
IN DAM.KNPD0 . SKIP TO TOP OF NEXT PAGE
IN DAM.KNPD1 . GET/CHECK PRINTER SPECIFICATIONS
IN DAM.KNDPOL . GET/CHECK POLAR TRANSFORMATION GAIN/BIAS
IN DAM.KNDSHA . GET/CHECK SHARPENING FILTER COEFFICIENTS
IN DAM.KNDSYM . GET/CHECK SYMBOLS
IN DAM.KNDTAB . TABULATE BY RADIANCE/SYM/SYMBOL/COLOR/INTENSITY

SEO 5-EXEC*. (S-READS,S-OPNCLPR,S-CALCROP)
IN DAM.KNDXX . MACRO COMMANDS
IN DAM.KND90 . DYNAMIC 90D
IN DAM.KND90 . DYNAMIC 90D
IN DAM.KND90 . DYNAMIC 90KPT
IN DAM.KND9F . DYNAMIC 9FREE
IN DSN.KDGOLO . DYNAMIC LOG

SEO 5-DISLISPIC. , (5-READS.5-OPNCLPR.5-CALCROP)
IN DSN.PICDI5 . DISPLAY RADIANCE/GRADIENT/CLASS (PHASE 0)
IN DSN.PICLI5 . LIST RADIANCE/GRADIENT/CLASS (PHASE 0)
IN DSN.PICPIC . PICTURE RADIANCE/GRADIENT/CLASS (PHASE 0)

SEO 5-FACROT*. (5-READS.5-OPNCLPR.5-CALCROP)
IN DSN.PICFAC . FACTOR CHANNELS (PHASE 0)
IN DSN.PICFAO . FACTOR CHANNELS (PHASE 0)
IN DSN.PICROT . ROTATE FACTOR STRUCTURE/COEFFICIENTS (PHASE 0)

SEO 5-PICPAR*. (5-READS.5-OPNCLPR.5-CALCROP)
IN DSN.PICPAR . PARTITION FACTOR SPACE (PHASE 0)

SEO 5-PICPRO*. (5-READS.5-OPNCLPR.5-CALCROP)
IN DSN.PICPRO . PROFILE RADIANCE (PHASE 0)

SEO 5-PICTOT*. (5-READS.5-OPNCLPR.5-CALCROP)
IN DSN.PICTOT . TOTAL TABULATIONS (PHASE 0)

. MONITOR FOR PHASE 3.4.5 COMMANDS -------------------------------

SEO 5-PIC345*. 5-PIC0129
IN DSN.PIC345 . CALL PREVIOUSLY NAMED PHASE 3/4/5 'TO' ROUTINE

. UTILITIES FOR PHASE 3.4.5 COMMANDS -------------------------------

SEO 5-RO3MP*. (5-PIC345)
IN DSN.RO3MP . READ MSS DATA IN OIL FORMAT (UNIT 3)
IN DSN.RO3MPQ . READ MSS DATA IN BSQ FORMAT (UNIT 3)

SEO 5-RO3BIP*. (5-PIC345)
IN DSN.RO3BIP . MSS DATA IN BIP FORMAT (UNIT 3)

SEO 5-RO3DSKNU*. (5-PIC345)
IN DSN.RO3DSK . DATA ON DISK IN PXBDEF FORMAT
IN DSN.RO3DSKU . SYNTHETIC DATA -- NO UNIT 3

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

SEO 5-PICDI3*. (5-RO3MP.5-RO3BIP.5-RO3DSKNU)
IN DSN.PICDI3 . DISPLAY RADIANCE
SEO 5-PICDI5*. (5-RO3MP.5-RO3BIP.5-RO3DSKNU)
IN DSN.PICDI5 . DISPLAY GRADIENT/LAPLACIAN/VARIANCE
SEO 5-PICDI5*. (5-RO3MP.5-RO3BIP.5-RO3DSKNU)
IN DSN.PICDI5 . DISPLAY CLASS

SEO 5-PICFA3*. (5-RO3MP.5-RO3BIP.5-RO3DSKNU)
IN DSN.PICFA3 . FACTOR CHANNELS

SEO 5-PICL13*. (5-RO3MP.5-RO3BIP.5-RO3DSKNU)
IN DSN.PICL13 . LIST RADIANCE
PICTA6-MAP

OAM PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

SEO 5-PICL14+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAN.PICL14 LIST GRADIENT/LAPLACIAN/VARIANCE
SEO 5-PICL15+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAN.PICL15 LIST CLASS

SEO 5-PICPA3+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAM.PICPA3 PARTITION BY DENSITY
SEO 5-PICPA4+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAM.PICPA4 PARTITION BY GRADIENT/LAPLACIAN/VARIANCE

SEO 5-PICP13+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAN.PICP13 PICTURE RADIANCE
SEO 5-PICP14+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAN.PICP14 PICTURE GRADIENT/LAPLACIAN/VARIANCE
SEO 5-PICP15+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAN.PICP15 PICTURE CLASS

SEO 5-PICPR3+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAM.PICPR3 PROFILE RADIANCE
SEO 5-PICT03+(S-RD3M0P.S-RD3B1P.S-RD3DSKNU)
IN OAM.PICT03 TOTAL TABULATIONS

MONITOR FOR PHASE 6.7.0 COMMANDS -----------------------------------------------
SEO 5-PIC676+(S-PICO01)
IN OAM.PIC676 CALL PREVIOUSLY NAMED PHASE 6/7/8 'TO' ROUTINE

PHASE 6.7.0 COMMANDS (NO FORTRAN I/O) --------------------------------------------
SEO 5-PICPA4+(S-PIC670)
IN OAM.PICPA4
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES
IN DAN.PICTAB/VIRTUAL
SUBROUTINE PIC0001  0 CALL PHASE 0 SUBROUTINES FOR PICTAB
I KASSUB)  0 NAME OF SUBROUTINE TO CALL (OR NULLSUB)

HISTORY
-------

E N SCHLOSSER  LEC  03/27/70  ORIgINAL CODE
E N SCHLOSSER  LEC  05/27/70  DELETE RETN K'S & NUMERIC OPTIONS
E N SCHLOSSER  LEC  01/19/70  MACRO COMMANDS & TIME COMMAND
J C CRISP  LEC  10/24/70  PEEK,POKE,COLOR,INTERN,PICTURE,IF,FI
J C CRISP  LENSCEO  05/18/80  CROSSTAB, CHANGE DISTAB TO KHDTab

METHOD
-------

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

MACHINE-DEPENDENT CODE
-------------------------
NONE.

EXTERNAL REFERENCES
---------------------
READS  0 READ PUNCHED CARD OR TERMINAL INPUT
GETAL  0 GET ALPHABETIC COMMAND
INTEGER ICE  0 INTEGER-CHAR-EQUV FOR CHARACTER
WARS  0 PRINT/LOG WARNING MESSAGE
PIC...  0 DEDICATED SUBROUTINE FOR COMMAND ... (SEE BELOW)
KNO...  0 COMMON SUBROUTINE FOR COMMAND ... (SEE BELOW)

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

GLOBAL DECLARATIONS
---------------------

LOCAL DECLARATIONS
--------------------
SUBROUTINE

CALL PREVIOUSLY NAMED SUBROUTINE
CALL TRACE
CALL NAMSUB  0 CALL TO NULSUB DOES NOTHING

READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)

IF(INBATCH.NE.0) CALL READS(LSTAT,  '1') 0 FILL BUFFER, BLANK CUE MSG
IF(INBATCH.EQ.0) CALL READS(LSTAT,  NULCST) 0 FILL BUFFER, NO CUE MSG
IF(LSTAT.NE. '1') KOND='COFS'

CALL GETSAL(KOND.(3),  NULCST) 0 GET 3 ALPHA CHARs

CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT

KASE=ICE(KOND)-ICE('A')+1  0 A TO Z = 1 TO 26

CASE STATEMENT ON MODIFIED 1-C-E OF COMMAND'S FIRST CHARACTER

IF(KASE.LT.11. OR. (KASE.GT.26)) KASE=8  0 NOT ALPHA
GO TO 101
0 901,402,403,404,405,406,407,408,409,410,
1 411,412,413,414,415,416,417,418,419,420,
2 421,422,423,424,425,426,427
5  ,KASE

DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOND TO BLANK

401 CONTINUE B**A
IF(KOND.EQ. 'ALT') CALL KMDALT(KOND) 0 ALIGN
GO TO 000

402 CONTINUE B**B
GO TO 000

403 CONTINUE B**C
IF(KOND.EQ. 'CHA') CALL KMDCHA(KOND) 0 CHANNEL
IF(KOND.EQ. 'CLE') CALL KMDCLE(KOND) 0 CLEAR
IF(KOND.EQ. 'COL') CALL KMDCOL(KOND) 0 COLOR
IF(KOND.EQ. 'COPI') CALL KMDCOPI(KOND) 0 COPIES
IF(KOND.EQ. 'CRO') CALL KMDCRO(KOND) 0 CROSSTAB
GO TO 000
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C 404 CONTINUE 8*** D
IF(KOMD.EQ.'DIS') CALL PICDIS(KOMD)
GO TO 800

C 405 CONTINUE 8*** E
IF(KOMD.EQ.'E05') CALL PICEX1(KOMD)
IF(KOMD.EQ.'EXI') CALL PICEX1(KOMD)
IF(KOMD.EQ.'EXP') CALL KMODEXP(KOMD)
GO TO 800

C 406 CONTINUE 8*** F
IF(KOMD.EQ.'FAC') CALL PICFAC(KOMD)
IF(KOMD.EQ.'FI') CALL KMODFI(KOMD)
GO TO 800

C 407 CONTINUE 8*** G
GO TO 900

C 408 CONTINUE 8*** H
IF(KOMD.EQ.'HEA') CALL KMODHEA(KOMD)
IF(KOMD.EQ.'HIS') CALL DISHIS(KOMD)
GO TO 800

C 409 CONTINUE 8*** I
IF(KOMD.EQ.'IF') CALL KMODIF1(KOMD)
IF(KOMD.EQ.'INT') CALL KMODINT(KOMD)
GO TO 800

C 410 CONTINUE 8*** J
C 411 CONTINUE 8*** K
GO TO 800

C 412 CONTINUE 8*** L
IF(KOMD.EQ.'LIN') CALL KMODLIN(KOMD)
IF(KOMD.EQ.'LIS') CALL PICLIS(KOMD)
GO TO 800

C 413 CONTINUE 8*** M
IF(KOMD.EQ.'MER') CALL KMODMER(KOMD)
GO TO 800

C 414 CONTINUE 8*** N
IF(KOMD.EQ.'NAM') CALL KMODNAM(KOMD)
IF(KOMD.EQ.'NEW') CALL KMODNEW(KOMD)
IF(KOMD.EQ.'NEX') CALL KMODNEXT(KOMD)
GO TO 800

C 415 CONTINUE 8*** O
IF(KOMD.EQ.'OFF') CALL KMODOFF(KOMD)
IF(KOMD.EQ.'ON') CALL KMODON(KOMD)
IF(KOMD.EQ.'ORI') CALL KMDDA(KOMD)
GO TO 800

C 416 CONTINUE 8*** P
IF(KOMD.EQ.'PAO') CALL KMODPAO(KOMD)
GO TO 800

GO TO 900
IF(KOMO.EQ.'PAR') CALL PICPAR(KOMO) 8 PARTITION
IF(KOMO.EQ.'PEE') CALL KMPPEE(KOMO) 8 PEEK
IF(KOMO.EQ.'PIC') CALL PICPIC(KOMO) 8 PICTURE
IF(KOMO.EQ.'POK') CALL KMPOK(KOMO) 8 POKE
IF(KOMO.EQ.'POL') CALL KMPOL(KOMO) 8 POLAR
IF(KOMO.EQ.'PRI') CALL KMPRI(KOMO) 8 PRINTER
IF(KOMO.EQ.'PRO') CALL PICPRO(KOMO) 8 PROFILE
GO TO 800

C 417 CONTINUE 8**** Q
     GO TO 800
C 418 CONTINUE 8**** R
     IF(KOMO.EQ.'RAD') CALL KMDRAD(KOMO) 8 RADIANCE
     IF(KOMO.EQ.'REN') CALL KMDREN(KOMO) 8 RENUMBER
     IF(KOMO.EQ.'ROT') CALL PICROT(KOMO) 8 ROTATE
     GO TO 800
C 419 CONTINUE 8**** S
     IF(KOMO.EQ.'SHA') CALL KMSHA(KOMO) 8 SHARPENING
     IF(KOMO.EQ.'SPA') CALL KMSPA(KOMO) 8 SPACING
     IF(KOMO.EQ.'SYM') CALL KMSYM(KOMO) 8 SYMBOLS
     GO TO 800
C 420 CONTINUE 8**** T
     IF(KOMO.EQ.'TAB') CALL KMTAB(KOMO) 8 TABULATE
     IF(KOMO.EQ.'TIC') CALL KM'TIC(KOMO) 8 TICS
     IF(KOMO.EQ.'TIM') CALL KMTIM(KOMO) 8 TIME
     IF(KOMO.EQ.'TOT') CALL PICTOT(KOMO) 8 TOTAL
     GO TO 800
C 421 CONTINUE 8**** U
C 422 CONTINUE 8**** V
     GO TO 800
C 423 CONTINUE 8**** W
     IF(KOMO.EQ.'WIN') CALL KMDWIN(KOMO) 8 WINDOW
     GO TO 800
C 424 CONTINUE 8**** X
C 425 CONTINUE 8**** Y
     GO TO 800
C 426 CONTINUE 8**** Z
     IF(KOMO.EQ.'ZON') CALL KMDZON(KOMO) 8 ZONE
     GO TO 800
C 427 CONTINUE 8**** NOT ALPHABETIC -- **ONLY FOR DEBUGGING**
     IF(KOMO.EQ.'BAD') CALL KMDBAD(KOMO) 8 BAD
     IF(KOMO.EQ.'SAD') CALL KMDSAD(KOMO) 8 SAD
     IF(KOMO.EQ.'SAS') CALL KMDSAS(KOMO) 8 SAS
     IF(KOMO.EQ.'BRK') CALL KMDBRK(KOMO) 8 BRKPT
     IF(KOMO.EQ.'SFR') CALL KMDSFR(KOMO) 8 SFREE
     IF(KOMO.EQ.'SLO') CALL KMDSLO(KOMO) 8 SLOO
     GO TO 800
C IF COMMAND WAS NOT FOUND, TRY MACRO-COMMAND
C
C 800 IF(KOMD.NE.'') KOMD='PIC' 8 1ST 3 CHAR OF PROG NAME PLUS ---
  IF(KOMD.NE.'') CALL KMOXXX(KOMD) 8 MACRO COMMAND HANDLER
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
C  IF(KOMD.NE.'') CALL WARNS('INVALID COMMAND ---')
C
C FORCE ALL FORTRAN I/O ROUTINES INTO SAME SEG AS PI0000 (NEVER PERFORMED)
C  IF(KOMD.EQ.'JUNK') READ(895,895) KOMD
  895 FORMAT(ix)
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
RETURN
END
SUBROUTINE PIC1291 & CALL PHASE 1/2/9 SUBROUTINES FOR PICTAB

1 NAMSUB1 & NAME OF SUBROUTINE TO CALL

C-------------------------------------------------------------

C
E N SCHLOSSER

C CALL PREVIOUSLY NAMED SUBROUTINE

CALL TRACE
CALL NAMSUB

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

RETURN
END
SUBROUTINE PIC349:  (CALL PHASE 3/4/5 SUBROUTINES FOR PICTAB
   NAMESUB)  B NAME OF SUBROUTINE TO CALL

C
C (E H SCHLOSSER)
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
C CALL TRACE
C CALL NAMESUB
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
C RETURN
C END
SUBROUTINE PIC0701  & CALL PHASE B/7/8 SUBROUTINES FOR PICTAB
I NAMSUB:  & NAME OF SUBROUTINE TO CALL

C-------------------------------------------------------------
C       C IC N SCNLOSSERI
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
CALL TRACE
CALL NAMSUB

C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
RETURN
END
SUBROUTINE PICDIS: 0 DISPLAY NSS-DERIVED DATA (PHASE 0)
U KONDO  & L: FIRST 3 CHARs OF COMMAND R: SPACES

HISTORY

E H SCHLOESSER LEC 10/05/75 ORIGinaL CODE
E H SCHLOESSER LEC 08/20/79 DELETE RETN K & ADD OPRPIC
E H SCHLOESSER LEC 03/13/79 SIMPLIFY LOGIC
J C CRISP LEC 11/06/79 REVISE FOR SPACING CHANGES

METHOD

CHECK/CALIBRATE SPECS. GENERATE TICK TABLE & DISPLAY HEADINGS.
THEN NAME PIC013/4/5 TO GENERATE BODY OF DISPLAY.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSKH: GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSN: GET INTEGER DATA FIELD FROM UNIT 5
MDWARN: PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
MDNOTE: PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
NVIAO: NAME 'VIA' 'TO' SUBROUTINES
CALCHA: CALIBRATE CHANNELS
CALSYN: CALIBRATE SYMBOL TABLE FOR OVERPRINTING
CALSPA: CALIBRATE TRANSFORMATION COEFFICIENTS FOR SPACING
CALWIN: CALIBRATE WINDOW ENVELOPES
CROPOM: CROP OUTPUT WINDOW
OPRPIC: OPEN ALTERNATE PRINT FILE(S)
ENTIC: GENERATE TICK TABLE
IDLU3: IDENTIFY LOGICAL UNIT 3
IDERT: IDENTIFY ERTS SCENE
IDCPIC: IDENTIFY CURRENT COMMAND SPECS FOR PICTAB
HDUNIT: WRITE HEADING LINE(S) AT TOP OF NEXT PAGE
WARN: SUBMIT WARNING FOR MISSING/INVALID FIELD FROM UNIT 5
EXTERNAL PIC000. NULSUB
EXTERNAL PIC129. PIC015
EXTERNAL PIC345. PIC013.PIC014.PIC015

EXCEPTIONS

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

<table>
<thead>
<tr>
<th>CONDITION</th>
<th>ACTION</th>
<th>DIAGNOSTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESSING DEFAULT COMMANDS</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>KLSTYP IN COMMON UNDEFINED</td>
<td>KLSTYP=&quot;RAD&quot;</td>
<td>NONE</td>
</tr>
<tr>
<td>KLSTYP SPECIFICATION MISSING</td>
<td>USE COMMON KLSTYP</td>
<td>NONE</td>
</tr>
<tr>
<td>KLSTYP SPECIFICATION INVALID</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>EXTRA SPECIFICATION</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>LIMIT CHANNEL VALUE RANGE IS NULL</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>SPACING &gt; 0.5 OR &lt; 1.9</td>
<td>DON'T GENERATE TICKS</td>
<td>NONE</td>
</tr>
<tr>
<td>NUMBER CONTROL POINTS &lt; 8</td>
<td>DON'T LIST TICKS</td>
<td>NONE</td>
</tr>
<tr>
<td>DEMAND RUN &amp; OVERPRINTED SYMBOLS</td>
<td>NO OVERPRINTING IN PICD13</td>
<td>NOTE</td>
</tr>
<tr>
<td>DATA/CHECKOUT MODE</td>
<td>'TO' ROUTINE IS NULSUB</td>
<td>NONE</td>
</tr>
<tr>
<td>WARNING(S) OR FATAL ERROR(S)</td>
<td>'TO' ROUTINE IS PICD19</td>
<td>NONE</td>
</tr>
</tbody>
</table>

GLOBAL DECLARATIONS

- INCLUDE KOMXGT.LIST  # COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
- INCLUDE KOMKLST.LIST  # COMMON CLASSIFICATION SPECTRAL LIMITS
- INCLUDE KOMFIT.LIST   # COMMON ADJUSTMENT/REGISTRATION PARAMETERS
- INCLUDE KOMTBL.LIST   # COMMON TICK/FREQ/FACTOR TABLE
- INCLUDE KOMSYM.LIST   # COMMON SYMBOL TABLE
- INCLUDE WINDEF.LIST   # DEFINE STRUCTURE OF WINDOW PACKETS
- INCLUDE KOMOWW.LIST   # COMMON OUTPUT WINDOW PACKETS
- INCLUDE NULCST.LIST   # DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

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

PROCEDURE

CALL TRACET

GET DISPLAY TYPE

KTLSTYP="NUL"  # MARK OLD FREQ TABLE AS DESTROYED
IF(NNNDOW.EQ.0) CALL HMWARNH ('INVALID DEFAULT COMMAND')
KLSAVE=KLSTPY  & SAVE PREVIOUS DISPLAY TYPE
IF(KLSTPY.NE.'GRA').AND.
& (KLSTPY.NE.'LAP').AND.
& (KLSTPY.NE.'VAR').AND.
& (KLSYPY.NE.'CLA'))
& IF UNDEFINED 
& THEN MAKE IT RADIANCE 
& CALL GETSRTY(KLSYPY,3), NULCST)
& UNLESS SPECIFIED BY DISPLAY CMD

C CHECK DISPLAY TYPE
C
IF(KLSYPY.NE.'RAD') GO TO 240  & RADIANCE?
 CALL NVIATO(PIC34S,PIC013) & NEXT CALL IS TO PIC013
 GO TO 300
240 IF(KLSYPY.NE.'GRA').AND.
& (KLSYPY.NE.'LAP').AND.
& (KLSYPY.NE.'VAR') GO TO 250  & VARIANCE?
 CALL NVIATO(PIC34S,PICD14) & NEXT CALL IS TO PICD14
 GO TO 300
250 IF(KLSYPY.NE.'CLA') GO TO 280  & CLASS?
 CALL NVIATO(PIC34S,PICD15) & NEXT CALL IS TO PICD15
 GO TO 300
280 CALL WARNH('BAD DISPLAY TYPE --')
KLSYPY=KLSAVE  & RESTORE PREVIOUS DISPLAY TYPE

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

C CHECK RADIANCE LIMITS
C
IF(LCVLOI.GT.LCVH11) CALL HMWARNH ('NO RADIANCE LIMITS')
IF(HDATAE.NE.0) GO TO 900  & DATA/CHECKOUT MODE

C CALIBRATE CHANNELS/SYMBOLS/SPACING/WINDOW
C
CALL CALCHA
CALL CALSYM
CALL CALSPA
CALL CALWINI 0.1

C OPEN PRINT FILE(S) IF NOT OPEN, CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
IF(NDOTL.NE.0) GO TO 900
IF(NNNDOW.LT.0) CALL ORRPIC  & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
NNNDOW=1485(NNNDOW)
NPAGE=0

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

L-65
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

NPROCIF=KPAGE-6  8 NUMBER OF PRINT COLUMNS PER FILE EXCLUDING
  8 LEFT MARGIN AND 8 HEAT LINE CHAR

CALL CROPW(I 1000,NALT+*NPROCIF)
  IF (NHTOTL NE 0) GO TO 900

C GENERATE TICKS (LIST ON UNIT 10 IF REGISTRATION ACCURATE)

LUNTIC=0
  IF (NCMPLT(I 9) LUNTIC=10  8 LIST TICKS ON UNIT 10
  IF (MSAOWW(WLIN,WSPI100).LT.50).OR.
    & (MSAOWW(WLIN,WSPI100).LT.1501) LUNTIC=-1  8 NO TICKS
  IF (MSAOWW(WSAM,WSPI100).LT.50).OR.
    & (MSAOWW(WSAM,WSPI100).LT.1501) LUNTIC=-1  8 NO TICKS
  CALL QENTIC(LUNTIC)

C COMPUTE SIZE OF PRINT WINDOW

NPRLIN=IFIX(PPOOWM(WLIN,WMAX))=IFIX(PPOOWM(WLIN,WMIN))+1
NPRLIN=IFIX(PPOOWM(WCOL,WMAX))=IFIX(PPOOWM(WCOL,WMIN))+1

C CHECK FOR DIAGNOSTICS

IF (NHTOTL NE 0) GO TO 900
  IF (MBATCH.EQ.0).AND. (NC1SYM.IE.1) CALL NDNOTE;
    & 'SYMBOLS NOT OVERPRINTED ON DEMAND TERMINAL'

C PRINT WINDOW HEADING FOR UNIT 6

WRITE(6,4191) NWNOOM,MTERAL
  419 FORMAT(' WINDOW NUMBER ',J=1,6X,'DISPLAY',6X,4A6)
  CALL IDLUS(6)
  CALL IDERT(6)
  CALL IDCPIC(6)

C PRINT WINDOW HEADING FOR ALTERNATE PRINT FILE(S)

LUNALT=10
  DO 480 N=1,NPROC,NPROCIF
    CALL MUNIT(4,LUNALT)
    WRITE(LUNALT,419) NWNOOM,MTERAL
    CALL IDLUS(LUNALT)
    CALL IDERT(LUNALT)
    CALL IDCPIC(LUNALT)
    LUNALT=LUNALT+1

480 CONTINUE

C ANY DIAGNOSTICS???

C 900 IF (NHTOTL.EQ.0) GO TO 900
  IF (MBATCH.NE.0) CALL NIATO(PICOSO.NULSUB) 8 DATA/CHECKOUT

L-87
IF(NDATAC.EQ.0) CALL NVTAT01

RETURN FOR CALL TO NAMED SUBROUTINE

RETURN
END
SUBROUTINE PIC013 1 DISPLAY RADIANCE (PHASE 2)

HISTORY

- E M SCHLOSSER  LEC  10/05/75  ORIGINAL CODE
- E M SCHLOSSER  LEC  06/27/76  UPGRADE DOCUMENTATION
- J C CRISP      LEC  11/08/79  REVISE FOR PBDEF FORMAT
- J C CRISP      LEMSCO  06/18/80  ADD 4 WORDS TO PRINT BUFFER

METHOD

- INITIALIZE LOW AND HIGH PRINT LINES AND COLUMNS. GENERATE AND PRINT TOP SAMPLE SCALE AND BORDER. INITIALIZE LOW AND HIGH ADJUSTED LINE. CALL OETRAD TO READ LINE. MASK NON-TRIVIAL WINDOW. RESAMPLE/SCREEN/COUNT FREQUENCY/SYMBOLIZE LINE. INSERT TICKS AND OUTPUT LINE. GENERATE AND PRINT BOTTOM SAMPLE SCALE AND BORDER. NAME PIC013 AS 'TO' ROUTINE FOR WRAP-UP OF DISPLAY PROCESSING.

MACHINE-DEPENDENT CODE

- UTILIZES UNIVAC EXEC 8 ER PRINTAS
  INTERNAL ROUTINE SAMSCC ASSUMES 8 CHAR TO AN INTEGER BIN

EXTERNAL REFERENCES

- AMP  0 ADJUSTED COORD FOR PRINT/LOT COORD
- OETRAD  0 GET ALL SELECTED RAM/TRANSFORMED CHANNELS
- MKSPIX  0 MASK NON-TRIVIAL WINDOW
- PENV1  0 PRINT/OVERPRINT FILES
- MODLTL  0 PRINT/LOG/COUNT 'FATAL ERROR' MESSAGES
- NVIATO  0 NAME 'VIA' 'TO' SUBROUTINES
- EM5RTA  0 WRITE TO ALTERNATE PRINT FILES
- CST4IN  0 CHARACTER STRING FOR INTEGER
- DOUBLE PRECISION COSWCS  0 VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH EXTERNAL PIC125. PIC013
- EXTERNAL RESSYN, GETBYT, GETVAL, GETINT, GETMUL  0 ROUTINE TO GET BIN VALUE

EXCEPTIONS

- STATUS FROM
  OETRAD  NEAT LINE CHAR  PRINT CELL SYMBOLS  DIAGNOSTIC ACTION
  'EOF'  1:1  NO DATA ('1:1')  NONE  PRINT LINE

L-69
GLOBAL DECLARATIONS

- INCLUDE KOMKXT.LIST 0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
- INCLUDE KOMKL5.LIST 0 COMMON CLASSIFICATION INFO
- INCLUDE KOMSYL.LIST 0 COMMON SYMBOL TABLE
- INCLUDE KINDEF.LIST 0 DEFINE STRUCTURE OF WINDOW PACKETS
- INCLUDE KOMONW.LIST 0 COMMON OUTPUT WINDOW PACKETS
- INCLUDE KONTBL.LIST 0 COMMON TICK/FREQ TABLE
- INCLUDE PIKDEF.LIST 0 DEFINE PICTAB PARAMETERS
- INCLUDE PCKDEF.LIST 0 DEFINE PIXEL BUFFER STRUCTURE
- INCLUDE MAXINT.LIST 0 MAXIMUM INTEGER VALUE

LOCAL DECLARATIONS

- PARAMETER NXBUF5-8 0 # OF MSS PIXEL BUFFERS IN ARRAY
- INTEGERS IN HSA BUF = #INTS PREAMBLE + (#INS-3)/4 + (#EXTRA BYTES-3)/4
- PARAMETER HMXBF = (PXBFNS-1) + (#INS-3)/4 + (#INS-3)/4
- WDS PRT BUF=WDS PREAM+(MAX FILES+MAX COLS/PO-HEIGHT COLS-HARC COLS)1
- PARAMETER HNPFIP=(PXBFNS-1)+INTLTHI+ (KPAGHI-2) + (4) + 4
- PARAMETER NFROCH-6 0 # WDS PER CHANNEL IN LOCAL FREQ TABLE
- PARAMETER NFROCFZ-120 0 # CHANNELS IN LOCAL FREQ TABLE
- INTEGER HPBF(NMXBF,NXBUF) 0 ARRAY OF MSS PIXEL BUFFERS
- INTEGER (PRTBF(NFROCF)) 0 PRINT BUFFER
- INTEGER NFROCNFROCF(0) 0 LOCAL FREQUENCY TABLE (SCOPE INCLUDES 8 INTERNAL ROUTINE RFSYN)
- INTEGER IPI10 0 PRINT LINE
- INTEGER ICPMN, ICPMAX 0 MINIMUM AND MAXIMUM PRINT COLUMN
- INTEGER IPI1MN, IPI1MAX 0 MINIMUM AND MAXIMUM PRINT LINE
- REAL ADJL1N,ADJSMN 0 ADJUSTED LINE AND SAMPLE
- INTEGER MLI00L,MLI00H,MLI00S 0 MSA LINE=100: LOW, HIGH, SPACING
- INTEGER MSALN 0 MSA LINE NUMBER
- INTEGER MSSHLO,MSASHI 0 LOW AND HIGH MSA SAMPLE
- INTEGER ISTAT 0 I/O STATUS
- INTEGER NTLCNR 0 NEAT LINE CHARACTER
- INTEGER NROLES,NROCS 0 LEFT & RIGHT MARGIN CHAR STRING
- INTEGER IPTLCN, IPTCNC, JSYTC 0 TICK PRINT LINE, COLUMN, SYMBOL
- INTEGER NPRLIN,NPROML 0 NUMBER OF PRINT LINES AND COLUMNS
- INTEGER LASTL0, LASTSCAN 0 LAST SCAN LINES READ

PROCEDURE

CALL TRACE
C INITIALIZE MINIMUM AND MAXIMUM PRINT LINES AND COLUMNS
C
  IPLMIN=PDDMIN(IPLMIN,PMIN)
  IPLMAX=PDDMAX(IPLMIN,PMAX)
  IPCMIN=PDDMIN(IPCOLUMN,CMIN)
  IPCMAX=PDDMAX(IPCOLUMN,CMAX)
C
C COMPUTE NUMBER OF PRINT LINES AND COLUMNS
C
  NPRMIN=IPLMAX-IPLMIN+1
  NPRCOL=IPCMAX-IPCMin+1
C
C INITIALIZE FIRST TICK
C
  CALL INITIC(PLTIC,IPCTIC,JSYTC)
C
C CLEAR LOCAL FREQUENCY TABLE
C
  DO 250 I=1,MLINCH
    DO 250 K=1,NFRQSZ
      NFREQ(K,1)=0
  250 CONTINUE
C
C INITIALIZE LOW AND HIGH LINES AND SPACING
C
  CALL ADP(AOJL,ADJSAM,FLOAT(IPLMIN),1.1)
  ML100L=ADJLIN+100.
  CALL ADP(AOJL,ADJSAM,FLOAT(IPLMAX),1.1)
  ML100H=ADJLIN+100.
  ML100S=HEADWIN(IPLMIN,WSPL100)
C
C POSITION AT TOP OF WINDOW
C
  MSALIN=ML100L/100
  CALL GETSPD(MPIBUFS,NBUFS,ISTAT,MSALIN.0,0)
  IF (ISTAT.NE.1) GO TO 300
    CALL NFATI ("BADF (BAD FILE) ON UNIT 3")
    CALL ERPTA ("102.6.*I/O ERROR - IGNORE OUTPUT")
  GO TO 300
C
C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE BEFORE 1ST PRINT LINE
C
  CALL SAMSCAL(IPRTBF,1(IPLMIN-1),1(IPCHM),(IPCHM))
  IF (1(WATCH.EQ.0).AND.(NPRMIN.LE.6).AND.(NPRCOL.LE.6))
    CALL PROVF1(I0,",",",",",",",",",","","","","",11111111",""10"",311000000"
    CALL PROVF1(I0,",",",",",","","",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",",","
C

IPLEIN=IPLEMIN
LASTLN=MAXINT
DO 350 NL100=NL100L,NL100H,NL100S
NLCHR=":"
MSALIN=NL100/100
CALL ANP (ADJLIN,ADJSAM,FLOAT(IPLIN),FLOAT(IPCHIN))
MSASLO=ADJSAM
CALL ANP (ADJLIN,ADJSAM,FLOAT(IPLIN),FLOAT(IPCMAX))
MSASHI=ADJSAM
IF (MSALIN.NE.LASTLN) CALL GETRAD (MPXBUF,NWXBF,NXBUF5).

ISTAT = MSALIN,MSASLO,MSASHI
LASTLN=MSALIN

IF (((ISTAT.EQ.'BADF') .AND. (ISTAT.NE.'OFL')) .OR. (PLCHIN.LE.64).AND.(NPRCOL.LE.64))
CALL M9FATL (CBS4CS(ISTAT,1,4)).

WHILE READING ON UNIT 3"
CALL ERPRTA('10'.2,6,'01/0 ERROR - IGNORE OUTPUT').

GO TO 900
320 IF (ISTAT.EQ.'BADF') NLCHR=":"
CALL MSKPIX (MPXBUF(1,1),MPXBUF(1,1))
CALL M9FATL (CBS4CS(ISTAT,1,4)).

GO TO 900
330 IF (MPXBUF(PX8INT,1).EQ.'BYT') CALL RESYM (IPRTBF.(IPLIN).
(IPCHIN),(IPCMAX),MPXBUF(NWXBF),(Noriously).GETBYT)
IF (MPXBUF(PX8INT,1).EQ.'CHR') CALL RESYM (IPRTBF,(IPLIN).
(IPCHIN),(IPCMAX),MPXBUF(NWXBF),(Noriously).GETICE)
IF (MPXBUF(PX8INT,1).EQ.'INT') CALL RESYM (IPRTBF,(IPLIN).
(IPCHIN),(IPCMAX),MPXBUF(NWXBF),(Noriously).GETINT)
IF (MPXBUF(PX8INT,1).EQ.'NUL') CALL RESYM (IPRTBF,(IPLIN).
(IPCHIN),(IPCMAX),MPXBUF(NWXBF),(Noriously).GETNUL)

CALL CST41N (MROLCS,(11),(6),MSALIN,4,'0')
HRORCS=MROLCS
IF ((MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64))
CALL PROVFI (10,MROLCS,4,'0',NLCHR,10000*,IPRTBF)
350 CONTINUE
C
C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE AFTER LAST PRINT LINE
C
CALL SAMSCL (IPRTBF,(IPMAX,1),(IPCHIN),(IPCMAX))
C
C MOVE DATA FROM LOCAL FREQUENCY TABLE TO COMMON TABLE (REPLACING TICKS)
C
L-72
C
C

C NEXT CALL IS TO PICD19
C
900 CALL NVIATO ( PIC129,PICD19)  
RETURN

C

C INTERNAL
SUBROUTINE INITIC  
& INITILIZE TICK POINTER AND GET FIRST TICK
0 IPLTIC,  & PRINT LINE FOR TICK
0 IPCTIC,  & PRINT COLUMN FOR TICK
0 JSYTC1  & TICK SYMBOL: '*'-PRIMARY, '.'-SECONDARY

GLOBAL DECLARATIONS

INCLUDE KOMTBL.LIST  & COMMON TICK TABLES AND FUNCTIONS

LOCAL DECLARATIONS

INTEGER JSYTC1(2)'/','.'/  & TICK SYMBOLS FOR TICK LEVELS 0 & 1
INTEGER NTICK  & TICK POINTER

PROCEDURE  

NTICK=0

ENTRY GETIC  
& GET NEXT TICK
0 IPLTIC,  & INTEGER PRINT LINE FOR TICK
0 IPCTIC,  & INTEGER PRINT COLUMN FOR TICK
0 JSYTC1  & TICK SYMBOL: '*'-PRIMARY, '.'-SECONDARY

PROCEDURE

C INCREMENT TICK POINTER


DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

\[ NTICK = NTICK + 1 \]

\[ \text{COMPUTE TICK LINE, COLUMN, LEVEL, AND SYMBOL} \]

\[ \text{IPLTIC} = \text{INTIC}(\text{NTICK}) \]
\[ \text{IPCTIC} = \text{COLTIC}(\text{NTICK}) \]
\[ \text{LVLTIC} = \text{LEVITIC}(\text{NTICK}) \]
\[ \text{JSYTIC} = \text{JSYB}(\text{LVLTIC} + 1) \]

\[ \text{RETURN} \]

\[ \text{INTERNAL} \]
\[ \text{SUBROUTINE SAMSCL} \]
\[ \text{GENERATE SAMPLE SCALE AND BORDER} \]
\[ \text{IPRTBF, A PRINT BUFFER} \]
\[ \text{IPRTBF, A PRINT BUFFER} \]
\[ \text{IPLIN, A PRINT LINE} \]
\[ \text{IPCMIN, A MINIMUM PRINT COLUMN} \]
\[ \text{IPCMAX, A MAXIMUM PRINT COLUMN} \]

\[ \text{METHOD} \]
\[ \text{INITIALIZE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER} \]
\[ \text{PREAMBLE, ENCODE SAMPLE NUMBERS AND PUT COLON, STRING, AND} \]
\[ \text{COLON IN PRINT BUFFER.} \]

\[ \text{MACHINE-DEPENDENT CODE} \]
\[ \text{ASSUMES 6 CHARS PER INTEGER BIN} \]

\[ \text{EXTERNAL REFERENCES} \]
\[ \text{AMP, A ADJUSTED COORD FOR PRINT/PLT COORD} \]
\[ \text{PUTCHR, A PUT CHAR IN CHAR STRING} \]
\[ \text{CST4IN, A CHARACTER STRING FOR INTEGER} \]

\[ \text{GLOBAL DECLARATIONS} \]
\[ \text{INCLUDE KOMOWW.LIST, A OUTPUT WINDOW PACKETS} \]
\[ \text{INCLUDE WINDSTRE.LIST, A DEFINE WINDOW PACKETS} \]
\[ \text{INCLUDE PXBDEF.LIST, A DEFINE BUFFER STRUCTURE} \]

\[ \text{LOCAL DECLARATIONS} \]
\[ \text{INTEGER IPRTBF(1)}, A \text{ARGUMENT} \]
C REAL ADJSAM  8 ADJUSTED SAMPLE
INTEGER IPBIN  8 POINTER TO PRINT BIN
INTEGER MSASAM  8 SAMPLE NUMBER
INTEGER MS100L,MS100H,MS100S  8 MSA SAMPLE+100; LOW,HIGH,SPACING

C C PROCEDURE
C C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C CALL ANP (ADJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCHIN))
MS100L=ADJSAM+100.
CALL ANP (ADJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCMAX))
MS100H=ADJSAM+100.
MS100S=MSAOWH(WSAM,WSPI00)

C C SET PREAMBLE POINTERS AND BIN POINTER
C IPRTBF(PXRECNI)=0
IPRTBF(PXLINO)=IPLIN
IPRTBF(PXCHAN)=0
IPRTBF(PXQUAL)=0
IPRTBF(PXINTI)="INT"
IPRTBF(PXLBIN)=1
IPRTBF(PXLCOL)=IPCHIN
IPRTBF(PXHBIN)=IPCMAX-IPCHIN+1
IPRTBF(PXH01)=PCMAX
IPRTBF(PXN01)=0
IPRTBF(PXNODE)=0
IPRTBF(PXR01)=0
IPRTBF(PXJ01)=0
IPBIN=IPRTBF(PXLBIN)-1

C C ENCODE SAMPLE NUMBER AND PUT COLON STRING AND COLON IN BUFFER
C DO 100 MS100L,MS100H,MS100S
MSASAM=MS100/100
CALL PUTCHR (IPRTBF(PXBIN)+IPBIN),(1),";"
CALL PUTCHR (IPRTBF(PXBIN)+IPBIN),(2),"",MSASAM,";0"
CALL PUTCHR (IPRTBF(PXBIN)+IPBIN),(3),";1";
IPBIN=IPBIN+1
100 CONTINUE

C C RETURN
INTERNAL SUBROUTINE RESYM  (A RESAMPLE/SCREEN/COUNT FREQUENCY/SYMBOLIZE/
A INSERT TICKS

0 IPRTBF. 8 PRINT BUFFER
1 IPLIN. 8 PRINT LINE
1 IPCMIN. 8 MINIMUM PRINT COLUMN
1 IPCMAX. 8 MAXIMUM PRINT COLUMN
2 I MPXBUF. 8 ARRAY OF MSS PIXEL BUFFERS
I NWIXBF. 8 NUMBER OF WORDS IN ONE BUFFER
( NXBUF'S. 8 NUMBER OF BUFFERS
I OETBINI 8 ROUTINE TO GET BIN VALUE--OETBYT.OETICE.OETINT.OETNUL

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 EXTERNAL REFERENCES
C
C ADJUSTED MSS COORD FOR PRINT/PLT COORD

C GLOBAL DECLARATIONS
C
C INCLUDE KOMOWN.LIST 8 COMMON OUTPUT WINDOW PACKETS
C INCLUDE KOMTBL.LIST 8 COMMON FRE/G TACK TABLE
C INCLUDE KOMKLS.LIST 8 COMMON CLASSIFICATION INFO
C INCLUDE KOMSYM.LIST 8 COMMON SYMBOL TABLE
C INCLUDE WINDEF.LIST 8 DEFINE WINDOW PACKETS

C LOCAL DECLARATIONS
C
PARAMETER NUM8FS •6 8 NUMBER OF MPX BUFFERS
INTEGER MPXBUF(NWIXBF.NXBUF'S) 8 ARGUMENT
INTEGER IPRTBF(11) 8 ARGUMENT
INTEGER NBINSO(NUM8FS) 8 BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER
INTEGER NBINCO 8 BIN NUMBER OF COLUMN 0 FOR PRINT BUFFER
INTEGER MSIOOL.MSIOOH.MS100S 8 MSA SAMPLE•100: LOW, HIGH, SPACING
INTEGER MSASAM 8 MSA SAMPLE NUMBER
INTEGER IPBIN 8 PRINT BUFFER BIN POINTER
INTEGER IPBITIC 8 PRINT BUFFER BIN FOR TICK
REAL ADJSAI 8 ADJUSTED SAMPLE NUMBER
INTEGER IPIXL1.IPIXL2.IPIXL3.
   & IPIXL4.IPIXL5.IPIXL6 8 PIXEL VALUE FOR EACH BUFFER

C
C PROCEDURE
C
C

L-78
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
CALL AUP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCHUN))
MS100L = ADJSAM = 100.
CALL AUP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCHUN))
MS100H = ADJSAM = 100.
MS100S = MSAGMN (MSAM, WSP100)

C set number of bin containing sample 0 for each MPX buffer
C
DO 150 NUMBUF = 1, NXBDFS
SBINSO(NUMBFF) = MPXBUF (PXLBIN, NUMBUF) - MPXBUF(PXLSAM, NUMBUF)
150 CONTINUE

C set preamble pointers and bin pointer
C
IPRTBF (PXRECN) = MPXBUF (PXRECN, 1)
IPRTBF (PXLIN0) = IPLIN
IPRTBF (PXCHAN) = 0
IPRTBF (PXQUAL) = 0
IPRTBF (PXINT) = 'INT'
IPRTBF (PXBLIN) = 2
IPRTBF (PXCL0L) = IPCMIN
IPRTBF (PXBIN) = IPCMAX - IPCMIN + 2
IPRTBF (PXCM) = IPCMAX
IPRTBF (PXNOIN) = 0
IPRTBF (PXNODA) = 0
IPRTBF (PXLOJ) = 0
IPRTBF (PXNOJ) = 0
IPRTBF (PXBI) = 0
NBIICO = IPRTBF (PXLBIN) - IPRTBF (PXCL0L) - NFORM INSERTING TICKS

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

C BUFFER 1
C
IF ((MSASAM.LT.MPXBUF(PXLSAM, 1)). OR. 
(MSASAM.GT.MPXBUF(PXHSAM, 1))) GO TO 350 
SAMPLE NOT IN BUFFER
CALL GETBIN (IPXLI, 
= MPXBUF(PXBIN, 1), MSASAM = NBIICO(1))
IF (IPXLI.GE.MPXBUF(PXNODA, 1)) GO TO 350 
NO DATA
IF (IPXLI.LT.LCVM11)) OR.
& 
(IPXLI.GT.LCVM11)) GO TO 360 
OUT OF RAD LIMITS
IF (NLMCH-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 (IPXLI, 

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

MPXBUF(PXBINS.2). (MSASAM+NBINS0(2)))
IF ((IP1XL2.LT.LCVLO(2)).OR.
(IP1XL2.GT.LCVHI(2))) GO TO 380  OUT OF RAD LIMITS
IF (NLIMCH.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  A SAMPLE NOT IN BUFFER
CALL GETBIN (IP1XL3.
MPXBUF(PXBINS.3). (MSASAM+NBINS0(3)))
IF ((IP1XL3.LT.LCVLO(3)).OR.
(IP1XL3.GT.LCVHI(3))) GO TO 380  OUT OF RAD LIMITS
IF (NLIMCH.3.EQ.0) GO TO 260

C BUFFER 4
C
IF ((MSASAM.LT.MPXBUF(PXLSAM.4)).OR.
(MSASAM.GT.MPXBUF(PXHSAM.4))) GO TO 350  A SAMPLE NOT IN BUFFER
CALL GETBIN (IP1XL4.
MPXBUF(PXBINS.4). (MSASAM+NBINS0(4)))
IF ((IP1XL4.LT.LCVLO(4)).OR.
(IP1XL4.GT.LCVHI(4))) GO TO 380  OUT OF RAD LIMITS
IF (NLIMCH.4.EQ.0) GO TO 230

C BUFFER 5
C
IF ((MSASAM.LT.MPXBUF(PXLSAM.5)).OR.
(MSASAM.GT.MPXBUF(PXHSAM.5))) GO TO 350  A SAMPLE NOT IN BUFFER
CALL GETBIN (IP1XL5.
MPXBUF(PXBINS.5). (MSASAM+NBINS0(5)))
IF ((IP1XL5.LT.LCVLO(5)).OR.
(IP1XL5.GT.LCVHI(5))) GO TO 380  OUT OF RAD LIMITS
IF (NLIMCH.5.EQ.0) GO TO 200

C BUFFER 6
C
IF ((MSASAM.LT.MPXBUF(PXLSAM.6)).OR.
(MSASAM.GT.MPXBUF(PXHSAM.6))) GO TO 350  A SAMPLE NOT IN BUFFER
CALL GETBIN (IP1XL6.
MPXBUF(PXBINS.6). (MSASAM+NBINS0(6)))
IF ((IP1XL6.LT.LCVLO(6)).OR.
(IP1XL6.GT.LCVHI(6))) GO TO 380  OUT OF RAD LIMITS

C COUNT FREQUENCY AND SYMBOLIZE
C
NFREQ(1PIXL6+1.6)+NFREQ(1PIXL6+1.6)+1
200 NFREQ(1PIXL5+1.5)+NFREQ(1PIXL5+1.5)+1
230 NFREQ(1PIXL4+1.4)+NFREQ(1PIXL4+1.4)+1
260 NFREQ(1PIXL3+1.3)+NFREQ(1PIXL3+1.3)+1
290 NFREQ(1PIXL2+1.2)+NFREQ(1PIXL2+1.2)+1
320 NFREQ(1PIXL1+1.1)+NFREQ(1PIXL1+1.1)+1
IPRTBF(PXBINS+IPBIN)=KSYM(IP1XL1+1)
GO TO 390
350 IPRTBF(PXBINS+IPBIN)=''  NO DATA SYMBOL
GO TO 380

L-78
C

C OCT AND INSERT TICKS

C

500  IPRTF(PXBINS+IPBIN)=""  SAVE TICK FOR SUBSEQUENT LINE
   IF (IPLTIC.GT.IPLIN) GO TO 900
   IF (IPRTBF(PXBINS+IPBIN).EQ.0) GO TO 530
      IF (IPLTIC.GT.IPLIN) THEN IONORE IT
      IF (JSYTIC.EQ.""') GO TO 530
         ALWAYS INSERT PRIMARY TICKS
         IF (IPRTBF(PXBINS+IPBIN).NE.""') GO TO 550
            IF (IPRTBF(PXBINS+IPBIN).NE.""') GO TO 550
               IF (IPRTBF(PXBINS+IPBIN).NE.""') HALO
                  IF (IPRTBF(PXBINS+IPBIN).NE.""') HALO
                     530 IPRTF(PXBINS+IPBIN)=JSYTIC
                     550 CALL GETIC (IPLTIC,IPCTIC,JSYTIC)
                     GO TO 500
   C

900 RETURN
END
SUBROUTINE PIC014 8 DISPLAY GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)

HISTORY

E H SCHLOSSER LEC 10/05/75 ORIGINAL CODE
E H SCHLOSSER LEC 10/18/79 UPGRADE DOCUMENTATION

NOT YET DESIGNED.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

MONOTE 3 PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGES
MONIATO 8 NAME 'VIA' 'TO' SUBROUTINES
DOUBLE PRECISION CBS4CS 8 VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH
EXTERNAL PIC129. "PIC019"

EXCEPTIONS

1. NOT YET IMPLEMENTED.

GLOBAL DECLARATIONS

INCLUDE KOMKLS.LIST 8 COMMON CLASSIFICATION INFO

LOCAL DECLARATIONS

NONE.

PROCEDURE

CALL TRACE

L-80
CALL MONOTE
- 'DISPLAY',
- CDBCS(KLSTYP.(1).(2))
- 'NOT YET IMPLEMENTED'
CALL NVIATO(PIC129,PICDIS)
RETURN
END
SUBROUTINE PICOIS & DISPLAY CLASS (PHASE 3)

(E N SCHLASSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

MONOTE
NVIATO

EXTERNAL PIC129,PICO19
CALL TRACE

CALL MONOTE('DISPLAY.CLASS NOT YET IMPLEMENTED')
CALL NVIATO(PIC129,PICO19)
RETURN
END
SUBROUTINE PIC019  & DISPLAY MBS-DERIVED DATA (PHASE 9)

**HISTORY**

- E H Schlosser  LEC  10/05/75  ORIGINAL CODE
- E H Schlosser  LEC  08/27/76  UPGRADE DOCUMENTATION
- E H Schlosser  LEC  05/01/79  DON'T INCREMENT WINDOW IF DIAGNOSTIC

**METHOD**

- CHECK DIAGNOSTIC COUNTERS AND PREPARE FOR NEXT DISPLAY.

**MACHINE-DEPENDENT CODE**

- NONE.

**EXTERNAL REFERENCES**

- **NOTE**
- **MDCLRN**

**EXCEPTIONS**

- NONE.

**GLOBAL DECLARATIONS**

- **INCLUDE KOMKDT.LIST**  8 COMMON PROGRAM EXECUTION SWITCHES,_counters
- **INCLUDE NULCST.LIST**  8 DEFINE NULL CHARACTER STRING
- **EXTERNAL PIC000.NULSUB**

**PROCEDURE**

- CALL TRACE

ON RETURN, CALL PIC000 TO GET COMMANDS

- CALL NVIATO(PICO00,NULSUB)
C ANY DIAGNOSTICS???

C 980 IF(NOFATL.EQ.0) GO TO 980
    CALL MONOTE('FATAL ERRORS -- NO DISPLAY GENERATED')
    GO TO 990

C 980 IF(NOMARK.EQ.0) GO TO 980
    CALL MONOTE('PREVIOUS MESSAGES -- NO DISPLAY GENERATED')
    IF(NMATCH.EQ.0) WRITE(8,989)
    FORMAT('X','**TRY AGAIN**')
    CALL MOCLM('NULCST')
    GO TO 990

C
C PREPARE FOR NEXT DISPLAY
C 960 MWNDON=MWNDON+1
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C 990 RETURN
END
SUBROUTINE DISHIS! 8 HISTOGRAM PREVIOUSLY DISPLAYED DATA
U (ONO) 8 I: FIRST 3 CHAR OF COMMAND 0: SPACES

(E H SCHLOSSER)

INCLUDE KORXOT.LIST
EXTERNAL PIC000.NULSUB

CALL NVIATO(PIC000.NULSUB)
CALL MONOTE('HISTOGRAM COMMAND NOT YET IMPLEMENTED')
KOMD
RETURN
END
SUBROUTINE PICEXI: A TERMINATION ROUTINE FOR PICTAB
U KOND1 8 1: FIRST 3 CHAR OF COMMAND 10: SPACES

METHOD
CLOSE/VERIFY INPUT ERTS TAPE. ASK ABOUT PRINTER DISPLAYS. PRINT
DISPLAYS ONSITE, IF REQUESTED. ELSE DELETE ALTERNATE PRINT FILES.
TERMINATE PROGRAM.

EXTERNAL REFERENCES
IOLU3  PRINT SHORT ID FOR LOGICAL UNIT 3
DELETE  DELETE ALTERNATE PRINT FILES
CLOSE3  CUSE AND VERIFY EOF ON INPUT ERTS TAPE
IDERTS  PRINT COMPLETE ERTS SCENE IDENTIFICATION
READS  FILL BUFFER FOR UNIT 5
CLOSPR  CLOSE ALTERNATE PRINT FILES
PSTOP  PROGRAM TERMINATION

EXCEPTIONS
NONE

GLOBAL DECLARATIONS
INCLUDE KOMXQT.LIST  COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST  COMMON ERTS SCENE PARAMETERS
INCLUDE NULCST.LIST  DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

L-96
INTEGER NCTY  // RESPONSE TO PRINT FILES ONSITE

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

C GET OPTIONAL 'QUICK' SPECIFICATION
   IF (NDATAC.NE.0) GO TO 900  // DATA/CHECKOUT MODE
   CALL GTSKKN(ITEMP(3), NULCST)
   IF (ITEMP.NE.'QUI') GO TO 120
   WRITE(6,105)
   105 FORMAT(4X,'***QUICK PROGRAM TERMINATION')
   GO TO 150

C CLOSE/VERIFY INPUT ERTS TAPE
   120 WRITE(6,125)
   125 FORMAT(4X,'***PROGRAM TERMINATION')
   CALL CLOSE3

C IDENTIFY ERTS SCENE WITH PITCH & ROLL IF JUST EXTRACTED FROM SIAT
   IF (INTNCT.NE.4).OR.(ABS(PITDEG).GT.9.) GO TO 150
     WRITE(6,145)
     145 FORMAT(1X)  // SKIP LINE
     CALL IOM331(6)
     CALL IOERTS(10)
     CALL IOLSU(10)

C BATCH RUN WITH FATAL ERRORS OR ANY DEMAND RUN -- ASK ABOUT PRINTER DISPLAY(S)
   IF ((MBATCH.NE.1).AND. (NDFATL.NE.0)) GO TO 200  // BATCH & OK
       IF (NDFATL.NE.0) CALL MDWARN('FATAL ERROR(S) ENCOUNTERED -- DISPLAY(S) ARE DEFECTIVE')
       WRITE(6,165)
     165 FORMAT(1X)  // OUTPUT DISPLAY(S) ON LINE PRINTER?'
       IF (KOMO.NE.'EOS') CALL READ5(L5STAT, 'I')
      IF (KOMO.NE.'Y') GO TO 700

C PRINT DISPLAY(S) ONSITE
   200 NWNOW=NNOW+MAX0(0,NNNOW-1)
   WRITE(6,245) NWNOW
245 FORMAT('DISPLAY(S) PRINTED')
CALL CLOSPR
GO TO 900

C
C DON'T PRINT DISPLAY(S) ONSITE
C 700 CALL QLETPR
C
C TERMINATE PROGRAM
C 900 CALL PSTOP("***PLEASE BFREE 3. OR BREWIND 3. OR BLOCATE 3.")
C
C PSTOP DOES NOT RETURN
C
END
SUBROUTINE PICFACT & FACTOR MSS CHANNELS (PHASE 0)
U KONO & 1: FIRST 3 CHARS OF COMMAND O: SPACES

METHOD
-------------------
CHECK COMMAND VALIDITY. CHECK FOR SPACING LESS THAN ONE. CALIBRATE SPACING/WINDOW AND CROP OUTPUT WINDOW. PRINT WINDOW HEADINGS AND NAME PICFA3 TO DO FACTORING.

MACHINE-DEPENDENT CODE
------------------------
NONE

EXTERNAL REFERENCES
---------------------
GETSIN & GET INTEGER DATA FIELD FROM UNIT 5
MOWARN & PRINT/LOG/COUNT 'WARNING' MESSAGES
CALCHA & CALIBRATE CHANNELS
CALSPA & CALIBRATE PRINT/PILOT COEFFICIENTS FOR SPACING
CALWIN & CALIBRATE WINDOW ENVELOPES
CROPOW & CROP MSS OUTPUT WINDOW TO FIT PRINT FILE SIZE
OPRPIC & OPEN ALTERNATE PRINT FILE
IDLU3 & PRINT SHORT ID FOR LOGICAL UNIT 3
IDENT & PRINT SHORT ERTS SCENE IDENTIFICATION
IDPIC & IDENTIFY CURRENT COMMAND SPECS
MVATO & NAME 'VIA' TO ROUTINES
EXTERNAL PIC129. PICFA9
EXTERNAL PIC345. PICFA3

EXCEPTIONS
-----------
1. SPACING OF LESS THAN ONE WILL GENERATE A WARNING DIAGNOSTIC AND NO FACTORING WILL BE PERFORMED.

GLOBAL DECLARATIONS
---------------------
INCLUDE KOMXQT.LIST & COMMON PROGRAM EXECUTION COUNTERS, SWITCHES
INCLUDE KONTBL.LIST & COMMON FACTOR TABLE
INCLUDE KOMNER.LIST & COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST & COMMON CLASSIFICATION INFO
INCLUDE KOMKRT.LIST & IRRADIANCE COEFFICIENT TRANSFORMS

L-89
C LOCAL DECLARATIONS
------------------
C
C	 NONE
C
C PROCEDURE
---------
CALL TRACE

C CHECK COMMAND VALIDITY
C IF (NWINDOW.EQ.0) CALL MOWARN('INVALID DEFAULT COMMAND')
C
C DRAIN SPECS FOR CURRENT COMMAND
C KLSTYP='RAD'  & DEFAULT IS FACTORING OF RADIANCE
CALL GETSIN(ITEMP+1.0,'EXTRA FACTOR SPECIFICATION --')
C
C CHECK FOR SPACING LESS THAN 1 -- (SPACING SCALED BY 100)
C IF (MSAOWW(WLIN,WSA100).GE.100).AND.
& (MSAOWW(WSAAM,WSA10000).GE.100)) GO TO 200
CALL MOWARN('REQUESTED SPACING LESS THAN 1.0')
GO TO 800
C
C CALIBRATE SPACING/CHANNELS/WINDOW
C 200 CALL CALSPA
CALL CALCMH
CALL CALWIN(0.1)
C
C CROP OUTPUT WINDOW TO FIT INPUT WINDOW
C CALL CROPOW(2500.3500)
C
C CHECK FOR DIAGNOSTICS
C IF (MDATA=/0) GO TO 800  & DATA/CHECKOUT MODE
C IF (NTOTL/.EQ.0) GO TO 800
C
C
C CLEAR WINDOW NUMBER
C
IF(MWNOOW.LT.01 CALL OPRPIC
  & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
  & NWNOOW=ABS(NWNOOW)
C
C PRINT WINDOW HEADING FOR UNIT 8
C
IF(MBATCH.NE.0) CALL HOUNIT(4,8)
  & WRITE(6,415) NWNOOW,MATERAL
  & 415 FORMAT(' WINDOW NUMBER ',J3,8X,'FACTOR',6X,4A5)
  & CALL IDLU3(8)
  & CALL IDERT(8)
  & CALL IDCPIC(8)
C
C PRINT WINDOW HEADING FOR UNIT 10
C
490 CALL HOUNIT(4,10)
  & CALL IDLU3(10)
  & CALL IDERT(10)
  & CALL IDCPIC(10)
C
C NAME SUBROUTINE TO 00 FACTORING
C
CALL NVIATO( PIC345,PICFA3)  & NEXT CALL IS TO PICFA3
  & 00 TO 900
C
C WARNINGS ENCOUNTERED -- NEXT CALL IS TO PICFA9
C
800 CALL NVIATO( PIC129,PICFA9)
  & KLSTYP=0  & NO FACTORING PERFORMED
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
900 KOMD=''
  & RETURN
  & END
SUBROUTINE PICFA3 & FACTOR RAW MSS CHANNELS (PHASE 3)

HISTORY

ML BROWN LEC 01/09/79 ALGORITHM CODING
JC CRISP LEC 10/09/79 REVISE INITIALIZATION OF CHANNELS AND FACTORS, CALL TO DSSPR3

METHOD

SET NUMBER OF CHANNELS AND FACTORS. COMPUTE SUMS AND SUMS OF PRODUCTS FOR MSS OR TEST DATA. COMPUTE CORRELATIONS, MEANS, STANDARD DEVIATIONS, EIGENVALUES, EIGENVECTORS, PERCENT OF VARIANCE FOR EACH FACTOR, FACTOR STRUCTURE, AND FACTOR COEFFICIENTS. MARK KTABLE AS CONTAINING FACTOR PARAMETERS.

MACHINE-DEPENDENT CODE

NONE

EXTERNAL REFERENCES

DSSPR A COMPUTE DBL PRECISION SUMS AND SUMS-OF-PRODUCTS
DCORLT A CORRELATIONS/MEANS/DVDS FROM D P SUMS/SUMS-OF-PRODS
JACMX A JACOBI ITERATION TO FIND E-VALUES/E-VECTORS
EISRTD A SORT E-VALUES/E-VECTORS BY DESCENDING E-VALUES
DOPCNT A COMPUTE % OF TRACE FOR MATRIX DIAGONAL ELEMENTS
DOSQRT A SQUARE ROOT OF MATRIX DIAGONAL ELEMENTS
DORVEC A RECIPROCAL OF MATRIX DIAGONAL ELEMENTS
MXMLT A MATRIX MULTIPLICATION
MDMARN A PRINT/LOG/COUNT 'WARNING' MESSAGES
NVIA TO NAME 'VIA' 'TO' SUBROUTINES
EXTERNAL PIC129, PICFA3

EXCEPTIONS

1. A WARNING WILL BE ISSUED IF A MAXIMUM NUMBER OF ITERATIONS IS EXCEEDED IN THE EIGENVALUE/EIGENVECTOR ROUTINE.
2. A WARNING WILL BE ISSUED IF THE NUMBER OF CHANNELS REQUESTED IS LESS THAN TWO.

GLOBAL DECLARATIONS

L-92
INCLUDE KOMXGT.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 CBS41N & VARIABLE-LENGTH CHAR STRING FOR INTEGER

C LOCAL DECLARATIONS
C---------------
PARAMETER NRCODIM=7
REAL TEST (10,3) / & 7...8...7...9...7...8...
& 4...3...5...2...3...5...4...2...
& 3...6...7...9...3...8...5...2/ & TEST DATA FROM COOLEY & LOMNES
(1971), p 110
DOUBLE PRECISION OSUM(NRCODIM) & SUMS
DOUBLE PRECISION OSPROD(NRCODIM,NRCODIM) & SUMS OF PRODUCTS
REAL EVALRR(NRCODIM,NRCODIM) & EIGENVALUES****5.5 (ON DIAGONAL)
INTEGER NROW,NCOL & ROW/COLUMN NUMBER

C PROCEDURE
C-----------
CALL TRACE
CALL KTBLTY=0

C FLAG KTBL AS DESTROYED

C INITIALIZE NUMBER OF CHANNELS AND FACTORS FOR MSS DATA
IF (LU3SEQ(1).EQ.'NUL') GO TO 180 & NO TAPE--USE TEST DATA
IF (NLIMCH.LT.1) CALL MOWARN ('ONLY ONE CHANNEL REQUESTED')
KCHACO=NLIMCH
KFACCO=KCHACO
IF (NOTOTL.NE.0) GO TO 900

C READ SCAN LINES--COMPUTE SUMS AND SUMS OF PRODUCTS
CALL OSSPR3 (OSUM,OSPROD,(NRCODIM),KPIXCO,KCHACO)
IF (KPIXCO .GE. 100) GO TO 200
CALL MOWARN ('PIXEL COUNT OF '.CBS41N(KPIXCO,3),' TOO SMALL')
GO TO 900

C COMPUTE NUMBER OF PIXELS, CHANNELS, FACTORS FOR TEST DATA
180 KPIXCO=10
KCHACO=3
KFACCO=3

L-93
COMPUTE SUMS & SUMS OF PRODUCTS FOR TEST DATA

CALL OSSPR(OSUM, OSPROD, (NRCDIM), TEST, KPIXCO, KCHACO)

COMPUTE CORRELATIONS, MEANS, STANDARD DEVIATIONS

CALL OCRRT(CORREL, CHMEAN, CHSTD.
* KPIXCO, DSUM, OSPROD, KCHACO, NRCDIM
DO 240 NSTO=1, KCHACO
  IF (CHSTD(NSTO) .GE. 1.) GO TO 240
    CALL MDWARN ('STD DEVIATION LESS THAN 1. FOR CHAN','
    CBS41(NSTO,2))
  GO TO 240
240 CONTINUE

DUPLICATE CORRELATION MATRIX

DO 280 NRW=1, KCHACO
  DO 280 NCW=1, KCHACO
    EIOVAL(NRW, NCW)=CORREL(NRW, NCW)
280 CONTINUE

COMPUTE EIGENVALUES & EIGENVECTORS & SORT IN ORDER OF DESCENDING EIGENVALUES

IT=50
CALL JACMX(EIOVAL, EIOVEC, NRCDIM, KCHACO, 5.E-7, IT, 5800.1)
CALL EISRTO(EIOVAL, EIOVEC, KCHACO, NRCDIM)

COMPUTE PERCENT OF VARIANCE ACCOUNTED FOR BY EACH FACTOR

CALL OOPCNMT(EIOVAL, KCHACO, KFACCO, NRCDIM, NRCDIM, PCTVAR)

COMPUTE FACTOR STRUCTURE (FSTRUCT)= [EIOVAL**.5] [EIOVEC]
IF (FSTRUCT) IS CORRELATION MATRIX BETWEEN CHANNELS (ROWS) & FACTORS (COLUMNS)

CALL DOSQRT(EIOVAL, KCHACO, KFACCO, NRCDIM, NRCDIM, EVALRR)
CALL MXMLT(EIOVEC, EVALRR, FSTRUCT, KCHACO, KFACCO, KCHACO, KFACCO, NRCDIM, NRCDIM)

COMPUTE NORMALIZED FACTOR COEFFICIENTS (FCNORM)= [EIOVAL**.5] [EIOVEC]

CALL DORECP(EVALRR, KCHACO, KFACCO, NRCDIM, NRCDIM, EVALRR) & RECIPROCAL
CALL MXMLT(EIOVEC, EVALRR, FCNORM, KCHACO, KCHACO, KFACCO, KFACCO, NRCDIM, NRCDIM)

MARK KTABLE AS CONTAINING FACTOR PARAMETERS FROM PRINCIPAL FACTOR ANALYSIS
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

KTLTY='FACT'
KTLNH=MIDDLE
GO TO 900

C
C FLAG ERROR DETECTED BY EIGENVALUE/EIGENVECTOR ROUTINE
C
800 CALL NDWARM('MAX ITERATIONS IN EIGEN')

C
900 CALL NVIATO(PIC128,PIC0)
C
RETURN
END
SUBROUTINE PICFAO 8 FACTOR RAW 1:55 CHANNELS (PHASE 2)

M L BROWN LEC 01/23/79 ALGORITHM CODING
J C CRISP LEC 12/08/79 UPGRADE DOCUMENTATION

METHOD
-----
PRINT RESULTS OF FACTORING FROM KTABLE.

MACHINE-DEPENDENT CODE
------------------------
NONE

EXTERNAL REFERENCES
---------------------
MONOTE
MVIAO
MATPRT
FACPR
EXTERNAL PICOOO.NULSUB

GLOBAL DECLARATIONS
---------------------
INCLUDE KOMXQT.LIST
INCLUDE KOMTBL.LIST

PROCEDURE
----------
CALL TRACE
CALL MVIAO(PICO00.NULSUB)

CHECK FOR DIAGNOSTICS

IF(KTBLTY.NE.'FACT') GO TO 800
IF(NOTOTL.NE.0) GO TO 800

125 FORMAT(' NUMBER OF PIXELS'/IX,112/1)
WRITE(8,125) KPIXCO

1'S FORMAT(' CHANNEL STANDARD DEVIATION2')
WRITE(*,142)
CALL HATPR(6,CHSTD,' 1.KFACCO.1.7)

289 FORMAT(' CHANNEL MEANS')
WRITE(*,289)
CALL HATPR(6,CHMEAN,' 1.KFACCO.1.7)

249 FORMAT(' CORRELATION BETWEEN CHANNELS')
WRITE(*,249)
CALL HATPR(6,CORREL,' 1.CHAN'.KCHACO.KFACCO.7.7)

329 FORMAT(' EIGENVALUES (DIAGONAL')
WRITE(*,329)
CALL HATPR(6,EIGVAL,' 1.ROW'.KCHACO.KFACCO.7.7)

349 FORMAT(' EIGENVECTORS')
WRITE(*,349)
CALL HATPR(6,EIVEC,' 1.ROW'.KCHACO.KFACCO.7.7)

389 FORMAT('OPRINCIPAL FACTOR ANALYSIS (UNROTATED) --')
WRITE(*,389)

399 FORMAT(' INDIVIDUAL (1) & CUMULATIVE (2) E OF VARIANCE BY FACTOR')
WRITE(*,399)
CALL HATPR(6,PCTVAR,' 1.2.KFACCO.2.7)

CALL FACPRF(FSTRUC.FCHNRM.FCORIG.CHSTD.CHMEAN.KCHACO.KFACCO.7.7)
GO TO 890

CHECK DIAGNOSTIC COUNTERS

800 IF(NNOWARN.EQ.0) GO TO 820
CALL HNOTE('PREVIOUS WARNINGS -- NO FACTORING PERFORMED')
IF(NBATCH.EQ.0) WRITE(*,819)
819 FORMAT(' ... TRY AGAIN')
GO TO 900

820 IF(NDFATL.EQ.0) GO TO 850
CALL HNOTE('PREVIOUS FATAL ERRORS -- NO FACTORING PERFORMED')
GO TO 900

850 IF(NCHECK.EQ.0) GO TO 890
CALL HNOTE('CHECKOUT MODE -- NO FACTORING PERFORMED')
GO TO 900

PREPARE FOR NEXT DISPLAY

890 MOWNDD=MOWNDD+1
C CLEAR MESSAGES & RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
900 CALL NCCLRM(-1)
RETURN
END
SUBROUTINE PICLIS: 0 LIST USS-DERIVED DATA (PHASE 01)
U KOND) 0% FIRST 3 CHAR OF COMMAND 01 SPACES

HISTORY

E N SCHLOSSER LEC 10/29/76 ORIGINAL CODE
O A BECK LEC 11/29/76 REVISE FOR SPACING CHANGES

METHOD

CHECK/CALIBRATE SPECS, GENERATE LIST HEADINGS.
THEN NAME PICLIS/9/5 TO GENERATE BODY OF LIST.

MACHINE-DEPENDENT CODE

USES UNIVAC FORTRAN V FUNCTION DOOL.

EXTERNAL REFERENCES

GETSBN 8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSBM 8 GET INTEGER DATA FIELD FROM UNIT 5
MODMRTN 8 PRINT/COUNT/LOG "WARNING" DIAGNOSTIC MESSAGE
MVIAOT 8 NAME 'VIA' 'TO' SUBROUTINES
CALSPA 8 CALIBRATE TRANSFORMATION COEFFICIENTS FOR SPACING
CALCHMA 8 CALIBRATE CHANNELS
CALWIN 8 CALIBRATE WINDOW ENVELOPES
CROPOH 8 CROP OUTPUT WINDOW
OPRPIC 8 OPEN ALTERNATE PRINTER FILE(S)
IDUS 8 IDENTIFY LOGICAL UNIT 3
IDENT 8 IDENTIFY ERTS SCENE
IDCPIC 8 IDENTIFY CURRENT COMMAND SPECS FOR PICTAB
HONUNT 8 WRITE HEADING LINE(S) AT TOP OF NEXT PAGE
WARN 0 SUBMIT WARNING FOR MISSING/INVALID FIELD FROM UNIT 5
EXTERNAL PICCOD. NULLSUB
EXTERNAL PIC19. PICL19
EXTERNAL PIC345. PICL3.PICL4.PICL5

EXCEPTIONS

1. 'LIST' MAY NOT BE A DEFAULT COMMAND.
2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE LIST.
3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS
<table>
<thead>
<tr>
<th>CONDITION</th>
<th>ACTION</th>
<th>DIAGNOSTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESSING DEFAULT COMMANDS</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>(NWNOOW=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLSTYP IN COMMON UNDEFINED</td>
<td>KLSTYP=’RAD’</td>
<td>NONE</td>
</tr>
<tr>
<td>KLSTYP SPECIFICATION MISSING</td>
<td>USE COMMON KLSTYP</td>
<td>NONE</td>
</tr>
<tr>
<td>KLSTYP SPECIFICATION INVALID</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>EXTRA SPECIFICATION</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>LIMIT CHANNEL VALUE RANGE IS NULL</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>(LCVLOL&gt;LCVHI)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA/CHECKOUT MODE</td>
<td>‘TO’ ROUTINE IS NULSUB</td>
<td>NONE</td>
</tr>
<tr>
<td>WARNING(S) OR FATAL ERROR(S)</td>
<td>‘TO’ ROUTINE IS PICLIS</td>
<td>NONE</td>
</tr>
</tbody>
</table>

C

LOCAL DECLARATIONS

<table>
<thead>
<tr>
<th>DECLARATION</th>
<th>USE</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDE KOMXT.LIST</td>
<td>a COMMON PROGRAM EXECUTION SWITCHES, COUNTERS</td>
</tr>
<tr>
<td>INCLUDE KOMKLS.LIST</td>
<td>a COMMON CLASSIFICATION SPECTRAL LIMITS</td>
</tr>
<tr>
<td>INCLUDE KOMTBL.LIST</td>
<td>a COMMON TICK/FREQ/FACTOR TABLE</td>
</tr>
<tr>
<td>INCLUDE WINDF.LIST</td>
<td>a DEFINE STRUCTURE OF WINDOW PACKETS</td>
</tr>
<tr>
<td>INCLUDE KOMOWN.LIST</td>
<td>a COMMON OUTPUT WINDOW PACKETS</td>
</tr>
<tr>
<td>INCLUDE NULCST.LIST</td>
<td>a DEFINE NULL CHARACTER STRING</td>
</tr>
</tbody>
</table>

C

PROCEDURE

---------

CALL TRACE

C

GET LIST TYPE

C

KSTBLTY=’ NUL’ | MARK OLD FREQ TABLE AS DESTROYED |
| IF(NWNOOW.EQ.0) CALL NWARN(‘INVALID DEFAULT COMMAND’) |
| KLSTYP=KLSTYP | SAVE PREVIOUS LIST TYPE |
| IF(KLSTYP.NE.’ORA’).AND. | |
| & (KLSTYP.NE.’LAP’).AND. | |
| & (KLSTYP.NE.’VAR’).AND. | |
| & (KLSTYP.NE.’CLA’)) | a IF UNDEFINED ...
| & KLSTYP=’RAD’ | a THEN MAKE IT RADIANCE) ...

L-100
CALL GETSKH(KLSTYP(3). NULCST) & ... UNLESS SPECIFIED BY LIST1 CMD

C C CHECK LIST TYPE
C
C IF(KLSTYP NE.'RAD') GO TO 240 & RADIANCE1?
CALL NVIATO( PIC345,PICL13) & NEXT CALL IS TO PICL13
GO TO 300
240 IF((KLSTYP NE.'GRA').AND.
& (KLSTYP NE.'LAP').AND.
& (KLSTYP NE.'VAR')) GO TO 250 & VARIANCE1?
CALL NVIATO( PIC345,PICL14) & NEXT CALL IS TO PICL14
GO TO 300
250 IF(KLSTYP NE.'CLA') GO TO 280 & CLASS1?
CALL NVIATO( PIC345,PICL15) & NEXT CALL IS TO PICL15
GO TO 300
280 CALL WARNS( 'BAD LIST TYPE '-' )
KLSTYP=KLSAVE & RESTORE PREVIOUS LIST TYPE
C
C DRAIN SPECS FOR CURRENT COMMAND
C
300 CALL GETSN(INTERN, +1.1.'EXTRA LIST SPECIFICATION --')
C
C C CHECK RADIANCE LIMITS
C
C IF(LCVLOI.OT.LCVHII) CALL MDWARN: 'NO RADIANCE LIMITS')
IF(MDATAC.NE.0) GO TO 900 & DATA/CHECKOUT MODE
C
C C CALIBRATE SPACING/CHANNELS
C
CALL CALSP/
CALL CALCHA
C
C C SAVE THE PHYSICAL PPD VERTEX COLUMNS
C
NODMAX=BOOL(.PPODW(HUSED.WHEAD))
DO 350 NOD+MW, NODMAX
PPODW(WCOL.NOD)=PPODW(WCOL.NOD)
350 CONTINUE
C
C C TRANSFORM TO LOGICAL PPD VERTEX COLUMNS FROM PHYSICAL
C
DO 375 NOD=MW, NODMAX
PPODW(WCOL.NOD)=PPODW(WCOL.NOD)/3
375 CONTINUE
C
C C CALIBRATE OUTPUT WINDOW ENVELOPE USING LOGICAL PPD VERTEX COLUMNS
C
CALL CALWIN( 0.)
C CROP OUTPUT WINDOW TO FIT INPUT WINDOW & ALT PRINT FILE(S)
C (3 PHYSICAL PPD COLUMNS PER LOGICAL PPD COLUMN)
C
NPRCIF=(KPAGE-4-21/3) 8 NUMBER OF LOGICAL PPD COLUMNS PER FILE
C 8 EXCLUDING LEFT MARGIN & 2 NEAT LINES
C CALL CROPWIN(1000,MALT*NPRCIF)
C
STORE THE PHYSICAL PPD VERTEX COLUMNS
C
DO 400 NODUW=NODMAX
PPDWIN(WCOL,NOD)=PPDWIN(WCOL,NOD)
400 CONTINUE
C
C CHECK FOR DIAGNOSTICS
C
IF (NOTOTL.NE.0) GO TO 900
C
C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
IF(NWNOOW.LT.0) CALL OPRPIC 3 OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
NWNOOW=ABS(NWNDOW) NPAGE=0
C
C COMPUTE LOGICAL SIZE OF PRINT WINDOW FROM LOGICAL PPD ENVELOPE
C
NPRLIN=1IFIX(PPDWIN(WLIN,WMAX))-1IFIX(PPDWIN(WLIN,WMIN))+1
NPRCOL=1IFIX(PPDWIN(WCOL,WMAX))-1IFIX(PPDWIN(WCOL,WMIN))+1
C
C CHECK FOR DIAGNOSTICS
C
IF (NOTOTL.NE.0) GO TO 900
C
C PRINT WINDOW HEADING FOR UNIT 6
C
WRITE(6,415) NWNOOW,MTERAL
415 FORMAT(' WINDOW NUMBER ',J3,6X,'LIST',8X,4A8)
CALL IDLUSI(6)
CALL IDERTI(6)
CALL I0CPICI(6)
C
C PRINT WINDOW HEADING FOR EACH ALTERNATE PRINT FILE IN LOGICAL PPD ENVELOPE
C
LUNALT=10
DO 460 N=1,NPRCOL,NPRCIF
CALL MUNIT(4,LUNALT)
WRITE(LUNALT,415) NWNOOW,MTERAL
CALL IDLUSI(LUNALT)
CALL IDERTI(LUNALT)
460CONTINUE

L-102
CALL IOCPIC( LUNALT)
LUNALT=LUNALT+1
480 CONTINUE

C
C ANY DIAGNOSTICS???
C
900 IF(INDTOTL.EQ.0) GO TO 990
   IF(MDATAEC.NE.0) CALL NVIATO( PIC000,NULSUB) 8 DATA/CHECKOUT
   IF(MDATAEC.EQ.0) CALL NVIATO( PIC129,PIC191)
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KOMD=''
   RETURN
END
SUBROUTINE PICL13 & LIST RADIANCE (PHASE 3)

HISTORY
-------
E W SCHLOSSER LEC 10/29/75 ORIGINAL CODE
D A BECK LEC 11/02/79 NEW PXBXEF BUFFER FORMAT
J C CRISP LEMSCO 08/19/80 ADD 4 WORDS TO PRINT BUFFER

METHOD
-------
INITIALIZE LOW AND HIGH PRINT LINES AND COLUMNS. GENERATE
AND PRINT TOP SAMPLE SCALE AND BORDER. INITIALIZE LOW AND
HIGH ADJUSTED LINE. CALL GETRAD TO READ LINE. MASK NON-
TRIVIAL WINDOW. RESAMPLE/SCREEN/COUNT FREQUENCY.
GENERATE AND PRINT BOTTOM SAMPLE SCALE AND BORDER.
NAME PICL19 AS 'TO' ROUTINE FOR WRAP-UP OF LIST
PROCESSING.

MACHINE-DEPENDENT CODE

EXTERNAL REFERENCES
---------------------
AMP & ADJUSTED COORD FOR PRINT/ PLOT COORD
GETRAD & GET ALL SELECTED RAW/TRANSFORMED CHANNELS
MKSPF & MASK NON-TRIVIAL WINDOW
PROVFI & PRINT/OVERPRINT FILES
MDSPATL & PRINT/LOG/COUNT 'FATAL ERROR' MESSAGES
NVIAO & NAME 'VIA' 'TO' SUBROUTINES
ERPRTA & WRITE TO ALTERNATE PRINT FILES
CSTMN & CHARACTER STRING FOR INTEGER
EXTERNAL PIC129, PICL19
EXTERNAL REPORT, GETBYT,GETICE,GETINT,GETNUL & ROUTINE TO GET BIN VALUE

EXCEPTIONS
----------
STATUS
FROM
GETRAD NEAT LINE CHAR PRINT CELL SYMBOLS DIAGNOSTIC ACTION
'EOF' ':' NO DATA ('::') NONE PRINT LINE

L-104
GLOBAL DECLARATIONS

INCLUDE KOMXQT.LIST  # COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST  # COMMON CLASSIFICATION INFO
INCLUDE WINDEF.LIST  # DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMKAW.LIST  # COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMLIB.LIST  # COMMON TICK/FREQ 'ABLE
INCLUDE PXBDEF.LIST  # DEFINE BUFFER STRUCTURE
INCLUDE PICODEF.LIST  # DEFINE PICTAB PARAMETERS
INCLUDE MAXINT.LIST  # MAXIMUM INTEGER VALUE

LOCAL DECLARATIONS

PARAMETER NXBUFS=6  # OF MSS PIXEL BUFFERS IN ARRAY

PARAMETER NMXBF=0  # MUA BUFFER PREAMBLE + (#BIN8+3)/4 + (#EXTRA BYTES+3)/4
PARAMETER NHXBF=(PIXINS-1) * (3548+314) + (19.314)

PARAMETER NWPIXBF=0  # MUA PIXEL BUFFER COUNT
PARAMETER NWPIXBF=(PIXINS-1) * (MALTHI * (KPAOH-2) - 11)  #

PARAMETER NFRQCH=5  # WDS PER CHANNEL IN LOCAL FREQ TABLE
PARAMETER NFRQSZ=128  # CHANNELS IN LOCAL FREQ TABLE

PARAMETER NFRQSZ=128  # CHANNELS IN LOCAL FREQ TABLE

INTEGER MPXBUF(NWXBF,NXBUFS)  # ARRAY OF MSS PIXEL BUFFERS
INTEGER IPRTBF(NWXBF)  # PRINT BUFFER

INTEGER NFREQ(NFRQSZ,NFRQCH)  # LOCAL FREQUENCY TABLE (SCOPE INCLUDES

INTEGER IPRINT  # PRINT LINE

INTEGER IPCHMIN,IPCHMAX  # MINIMUM, MAXIMUM PRINT COLUMN
REAL ADJLIN,ADJSAM  # ADJUSTED LINE AND SAMPLE

INTEGER ML100D,ML100H,ML100S  # MSA LINE#100: LOW,HIGH,SPACING

INTEGER MSALEN  # MSA LINE NUMBER

INTEGER ISTAT  # I/O STATUS

INTEGER NLINCH  # NEAT LINE CHARACTER

INTEGER MROLCS,MRORCS  # LEFT & RIGHT MARGIN CHAR STRING

INTEGER IPRINT  # PRINT COLUMN

INTEGER ML100  # MSA LINE#100

INTEGER MSA1LO,MSASHI  # LOW & HIGH MSA SAMPLE NUMBER

INTEGER DPRIN,DP1COL  # PRINT LINES, COLUMNS

INTEGER LASTLN  # LAST SCAN LINE READ

INTEGER I.K  # DO LOOP INDEXES

PROCEDURE

CALL TRACE

L-109
C INITIALIZE MINIMUM AND MAXIMUM PRINT LINES AND COLUMNS
C
IPLMIN=PDDOM(WLIN.WMIN)
IPLMAX=PDDOM(WLIN.WMAX)
IPCHMIN=PDDOM(WCOL.WMIN)
IPCHMAX=PDDOM(WCOL.WMAX)

C CALCULATE THE NUMBER OF PRINT LINES, COLUMNS TO BE PRINTED
C
NPRLIN=IPLMAX-IPLMIN+1
NPRCOL=(IPCHMAX-IPCHMIN+1)*3

C CLEAR LOCAL FREQUENCY TABLE
C
DO 250 I=1,NLIMCH
250 CONTINUE
DO 200 K=1,NFREQS
NFREQS(K,1)=0
200 CONTINUE

C INITIALIZE LOW AND HIGH LINES AND SPACING
C
CALL AVP (AOJLIN,ADJSAM, FLOAT(IPLMIN),1.)
ML100L=ADJLIN*100.
CALL AVP (AOJLIN,ADJSAM, FLOAT(IPLMAX),1.)
ML100H=ADJLIN*100.
ML100S=MSAOWN(WLIN.WSP100)

C POSITION AT TOP OF WINDOW
C
MSALIN=ML100L/100
CALL GTRAD (MPXBUF.(0),(NXBUFSI.INTSTAT, MSALIN,0.0)
IF (1STAT.NE.'BADF') GO TO 300
CALL MDFATL ('BADF (BAD FILE) ON UNIT 3')
CALL ERPTA (')
& '10 '.2,NWNC(28),'.I/O ERROR -- IGNORE OUTPUT')
GO TO 900

C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE BEFORE 1ST PRINT LINE
C
300 CALL SAMSCL (IPRTBF,(IPLMIN-1),(IPCHMIN),(IPCHMAX))
IF (MBATCH.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRCOL.LE.64)
& CALL PROVFI (B,'*',4,'*',0,','*1111',IPRTBF)
CALL PROVFI (10,,'*',4,'*',4,'*',4,'*',11111',IPRTBF)

C READ, MASK, RESAMPLE, SCREEN, COUNT FREQUENCY AND PRINT SCAN LINES
C
IPLIN=IPLMIN
LASTLN=MAXINT

L-108
GO 350 MLI00=ML100/100
MSALIN=ML100/100
CALL AVP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCHIN))
MSASL0=ADJSAM
CALL AVP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCHAX))
MSASHI=ADJSAM
IF (MSALIN.NE.LASTLN) CALL OETRAD (MPXBUF, NWIXBF, NXBUFS).
LASTLN=MSALIN
IF ((ISTAT.NE.'BADF').AND.((ISTAT.NE.'OFL')) GO TO 320
CALL MDFAFL (CBS4CS(ISTAT,1,4)).
WHILE READING ON UNIT 3'
CALL ERRTA (10, '2', '41/0 ERROR - IGNORE OUTPUT').
GO TO 900
320 IF (ISTAT.EQ.'BADR') NTLCHR='?'
CALL MXPIX (MPXBUF(1,1), MPXBUF(1,1)).
GO TO 900
330 IF (MPXBUF(PXBINT,1).EQ. 'BYT') CALL RESCRN (IPRTBF, IPLIN).
IF (MPXBUF(PXBINT,1).EQ. 'CHR') CALL RESCRN (IPRTBF, IPLIN).
IF (MPXBUF(PXBINT,1).EQ. 'INT') CALL RESCRN (IPRTBF, IPLIN).
IF (MPXBUF(PXBINT,1).EQ. 'NUL') CALL RESCRN (IPRTBF, IPLIN).
CALL CSTYIN (MROLCS, (11, MROLCS, 4, 'O').
MROLCS=MROLCS
IF (MBATCH, AND. (NPRLIN.LE.64), AND. (NPRCOL.LE.64))
CALL PROVFI (6, ROLLCS, 4, 'o', NTLCHR, '****', IPRBF)
CALL PROVFI (10, ROLLCS, 4, '****', '****', '****', IPRBF)
350 CONTINUE

C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE AFTER LAST PRINT LINE
C
C CALL SAMSC1 (IPRTBF, IPLMAX+1, IPCHIN)
C IF (MID BATCH, AND. (NPRLIN.LE.64), AND. (NPRCOL.LE.64))
C CALL PROVFI (6, '****', '****', '****', '****', '****', IPRBF)
C CALL PROVFI (10, '****', '****', '****', '****', '****', IPRBF)

C MOVE DATA FROM LOCAL FREQ TABLE TO COMMON TABLE
C
KTBLTY='FREQ'
KTBLNM=NWNDOW
GO 600 1=1, NL1MCH
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

0050 K=1,NFREQZ
KFREQ(K,1)=NFRGK(K,1)
0050 CONTINUE
0060 CONTINUE

C

C NEXT CALL IS TO PICL19
C
0090 CALL NYIATO ( PIC129,PICL19)
RETURN

C

C INTERNAL
SUBROUTINE SAMSCL( I GENERATE SAMPLE SCALE AND BORDER
0 IPRTBF, I PRINT BUFFER
1 IPLIN, I PRINT LINE
0 IPCMIN, A MINIMUM PRINT COLUMN (LOGICAL)
0 IPCMAX, A MAXIMUM PRINT COLUMN (LOGICAL)
C

C HISTORY
C
O A BECK LEC 11/07/79 REQUIREMENTS
C O A BECK LEC 11/07/79 ALGORITHM DESIGN
C O A BECK LEC 11/07/79 ALGORITHM CODING
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.
NOTES, 3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN.

C MACHINE-DEPENDENT CODE
C
ASSUMES 8 CHARACTERS PER INTEGER BIN

C EXTERNAL REFERENCES
C
AUX A ADJUSTED COORD FOR PRINT/PLOT COORD
C PUTCHR A PUT CHAR IN CHAR STRING
C CST4IN A CHARACTER STRING FOR INTEGER
C

C EXCEPTIONS
C
ASSUMES 8 CHARACTERS PER INTEGER BIN.
GLOBAL DECLARATIONS

INCLUDE KOMOWW.LIST
INCLUDE WINDF.LIST
INCLUDE PXBDEF.LIST

LOCAL DECLARATIONS

INTEGER IPRTB(1) 8 ARGUMENT
REAL ADJLIN, ADJSAM 8 ADJUSTED LINE, SAMPLE
INTEGER IPBIN 8 POINTER TO PRINT BIN
INTEGER MSASAM 8 SAMPLE NUMBER
INTEGER MS100L, MS100H, MS100S 8 MSA SAMPLE*100: LOW, HIGH, SPACING

PROCEDURE

INITIALIZE LOW AND HIGH SAMPLES AND SPACING

CALL AMP(ADJLIN, ADJSAM, FLOAT(IPLIN1, FLOAT(IPCHIN)))
MS100L = ADJSAM + 100.
CALL AMP(ADJLIN, ADJSAM, FLOAT(IPLIN1, FLOAT(IPCHAX)))
MS100H = ADJSAM + 100.
MS100S = MSADWM(MSAM, WP100)

SET PREAMBLE POINTERS

IPRTBF(PXRECH) = 0
IPRTBF(PXLINQ) = IPLIN
IPRTBF(PXCHAN) = 0
IPRTBF(PXQUAL) = 0
IPRTBF(PXBINT) = 1
IPRTBF(PXLBIN) = 2
IPRTBF(PXLCOL) = IPCLIN

CONVERT LOGICAL PPD COLUMNS TO PHYSICAL PPD BINS

IPRTBF(PXHBIN) = ((IPCHAX-IPCHIN+1)*3) + 1
IPRTBF(PXHCOL) = IPCHAX
IPRTBF(PXQIN) = IPCHAX
IPRTBF(PXQODA) = 0
IPRTBF(PXJIN) = 0
IPRTBF(PXJO1) = 0
IPRTBF(PXHJ0) = 0

INITIALIZE BIN NUMBER TO BIN ONE

IPBIN = IPRTB(PXLBIN) - 1

BUILD SCALE AND BORDERS

(3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN)
C

DO 100 M9100=MS100/100
     MSASAM=MS100/100
     IPRTBF(IPXSINS+IPBIN+0)=' ',
     IPRTBF(IPXSINS+IPBIN+1)='
     CALL PUTCHR(IPRTBF(IPXSINS+1*IPBIN+2),•(1),••'
     CALL CST4IN(IPRTBF(IPXSINS+IPBIN+2),21.4,MSASAM.4.'G') 8:9999:
     CALL PUTCHR(IPRTBF(IPXSINS+IPBIN+2),•(0),1
   100 CONTINUE

C
C RETURN TO CALLING ROUTINE
C
900 RETURN
C
C
C
C INTERNAL
C
SUBROUTINE RESCRN( I RESAMPLE/SCREEN/COUNT FREQUENCY
C IPRTBF, I PRINT BUFFER
C IPLIN. I PRINT LINE
C IPCHIN. I MINIMUM PRINT COLUMN (LOGICAL)
C IPCHMAX. I MAXIMUM PRINT COLUMN (LOGICAL)
C MPXBUF. I ARRAY OF MSS PIXEL BUFFERS
C NWIXBF. I NUMBER OF WORDS IN ONE BUFFER
C NXBIFS. I NUMBER OF BUFFERS
C GETBIN) I ROUTINE TO GET BIN VALUE--GETBYT.GETICE.GETINT.GETNUL
C
C HISTORY
C
D A BECK	LEC 11/07/79	REQUIREMENTS
D A BECK	LEC 11/07/79	ALGORITHM DESIGN
D A BECK	LEC 11/07/79	ALGORITHM CODING
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 MACHINE-DEPENDENT CODE
C
NONE.

L-110
C

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

C EXTERNAL REFERENCES

C
C AMP  0 ADJUSTED MSS COORD FOR PRINT/PLAT COORD
C CSTWIN 0 CHARACTER STRING FOR INTEGER
C GETCHR 0 GET CHARACTER FROM CHARACTER STRING

C EXCEPTIONS

C
C NONE.

C GLOBAL DECLARATIONS

C
C INCLUDE KONWM.LIST 0 COMMON OUTPUT WINDOW PACKETS
C INCLUDE KOMKL.LIST 0 COMMON FREQ/TICK TABLE
C INCLUDE KOMKL.LIST 0 COMMON CLASSIFICATION INFO
C INCLUDE PXBBDEF.LIST 0 DEFINE BUFFER STRUCTURE
C INCLUDE KOMSYN.LIST 0 COMMON SYMBOL TABLE
C INCLUDE WINDEF.LIST 0 DEFINE STRUCTURE OF WINDOW PACKETS

C LOCAL DECLARATIONS

C
INTEGER M
-NEXTFINWXI.FNXSUFS) 0 ARGUMENT
INTEGER IPRTBF(1) 0 ARGUMENT
INTEGER IPXL1,IPXL2,IPXL3,IPXL4,IPXL5,IPXL6 0 VALUE OF PIXEL FOR
EACH MPX BUFFER
PARAMETER NUMBF=0
INTEGER NSINSO(NUMBS) 0 BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER
INTEGER MS100L,MS100H,MS100S 0 MSA SAMPLE*100: LOW,HIGH,SPACING
INTEGER MSASAM 0 MSA SAMPLE NUMBER
INTEGER IPBIN 0 PRINT BUFFER BIN POINTER
REAL AJCLIN,ADJSAM 0 ADJUSTED LINE AND SAMPLE
INTEGER MS100 0 MSA SAMPLE+100
INTEGER KSTPIX 0 CHARACTER STRING OF PIXEL RADIANCE

C

C PROCEDURE

C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
CALL AMP (AJCLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCHIN))
MS100L=ADJSAM+100.
CALL AMP (AJCLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCHMAX))
MS100H=ADJSAM+100.
MS100S=MSADWMSAM,MSAM,MS100)

C
C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C
DO 150 NUMBF=1,NXBUF
NSINSO(NUMBUF)+=MPXBUF(IPXBIN,NUMBUF)-MPXBUF(IPXSAM,NUMBUF)
150 CONTINUE

C
C SET PREAMBLE POINTERS AND BIN POINTER
C
IPRT0F(PXRECH)=MPXBUF(PXRECH,1)
IPRT0F(PXKINO)=IPLIN
IPRT0F(PXCHAN)=0
IPRT0F(PXQUAL)=0
IPRT0F(PXBIN)="INT"
IPRT0F(PXLBIN)=0
IPRT0F(PXLCOL)=IPCHIN

C CONVERT LOGICAL PPD COLUMNS TO PHYSICAL PPD BINS
C
IPRT0F(PXBIN)=(1(IPCHAX-IPCHIN+1)*31+1)

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.MPXBUP(PXLSAM,1)) .OR. 
  (MSASAM.GT.MPXBUP(PXLSAM,1))) GO TO 350  A SAMPLE NOT IN BUFFER
CALL GETBIN (IPIXL1).
IF (IPIXL1.LT.MPXBUP(PXHODA,1)) . GO TO 350  A NO DATA
IF (IPIXL1.LT.LCVLO(1)) .GO TO 360  OUT OF RAD LIMITS
 IF (NLIMCH-1.EQ.0) GO TO 320
C
C BUFFER 2
C IF ((MSASAM.LT.MPXBUP(PXLSAM,2)) .OR. 
  (MSASAM.GT.MPXBUP(PXLSAM,2))) GO TO 350  A SAMPLE NOT IN BUFFER
CALL GETBIN (IPIXL2).
IF (IPIXL2.LT.MPXBUP(PXHODA,2)) . GO TO 350  A NO DATA
IF (IPIXL2.LT.LCVLO(2)) .GO TO 363  OUT OF RAD LIMITS
 IF (NLIMCH-2.EQ.0) GO TO 290
C
C BUFFER 3
C IF ((MSASAM.LT.MPXBUP(PXLSAM,3)) .OR. 
  (MSASAM.GT.MPXBUP(PXLSAM,3))) GO TO 350  A SAMPLE NOT IN BUFFER
CALL GETBIN (IPIXL3).
IF (IPIXL3.LT.MPXBUP(PXHODA,3)) . GO TO 350  A NO DATA
IF (IPIXL3.LT.LCVLO(3)) .GO TO 365  OUT OF RAD LIMITS

L-112
AN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IF (NLINCH-3.EQ.0) GO TO 260
C
C BUFFER 4
C
IF ((HSASAM.LT.HPXBUF(PXLSAM,9)).OR.
  (HSASAM.GT.HPXBUF(PXLSAM,9))) GO TO 350 \ SAMPLE NOT IN BUFFER
  CALL GETBIN (IPIXLS)
  MXPBUF(IPXBSN,9),(HSASAM-NSBSNO(9))
  IF ((IPIXLS.GT.LCVL0(9)).OR.
   (IPIXLS.GT.LCVL0(9))) GO TO 360 \ OUT OF RAD LIMITS
   IF (NLINCH-4.EQ.0) GO TO 230
C
C BUFFER 5
C
IF ((HSASAM.LT.HPXBUF(PXLSAM,10)).OR.
  (HSASAM.GT.HPXBUF(PXLSAM,10))) GO TO 350 \ SAMPLE NOT IN BUFFER
  CALL GETBIN (IPIXLS)
  MXPBUF(IPXBSN,10),(HSASAM-NSBSNO(10))
  IF ((IPIXLS.GT.LCVL0(10)).OR.
   (IPIXLS.GT.LCVL0(10))) GO TO 360 \ OUT OF RAD LIMITS
   IF (NLINCH-5.EQ.0) GO TO 220
C
C BUFFER 6
C
IF ((HSASAM.LT.HPXBUF(PXLSAM,11)).OR.
  (HSASAM.GT.HPXBUF(PXLSAM,11))) GO TO 350 \ SAMPLE NOT IN BUFFER
  CALL GETBIN (IPIXLS)
  MXPBUF(IPXBSN,11),(HSASAM-NSBSNO(11))
  IF ((IPIXLS.GT.LCVL0(11)).OR.
   (IPIXLS.GT.LCVL0(11))) GO TO 360 \ OUT OF RAD LIMITS
C
C COUNT FREQUENCY
C
NREQ(IPIXLS+1.6)+NREQ(IPIXLS-1.6)+1
200 NREQ(IPIXLS+1.5)+NREQ(IPIXLS-1.5)+1
230 NREQ(IPIXLS+1.4)+NREQ(IPIXLS-1.4)+1
260 NREQ(IPIXLS+1.3)+NREQ(IPIXLS-1.3)+1
290 NREQ(IPIXLS+1.2)+NREQ(IPIXLS-1.2)+1
320 NREQ(IPIXLS+1.1)+NREQ(IPIXLS-1.1)+1
C
C INSERT RADIANCE VALUE INTO PRINT BUFFER
C
(3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN)
C
CALL CST4(IN(KSTPIX.,(1)).)
  1PIXL1.3,'0')
  CALL GETCHR(IPRTBF(IPXBSN+IPRN=0)) \ HUNDREDS DIGIT
  KSTPIX.,(11)
  CALL GETCHR(IPRTBF(IPXBSN+IPBIN=1)) \ TENS DIGIT
  KSTPIX.,(12)
  CALL GETCHR(IPRTBF(IPXBSN+IPBIN=2)) \ UNITS DIGIT
  KSTPIX.,(13)
  IF((IPRTBF(IPXBSN+IPBIN=0).EQ.'0') IPRTBF(IPXBSN+IPBIN=0)):
   GO TO 290
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 INSERT 'NO INFO' SYMBOLS INTO PRINT BUFFER
C
360  IPRTBF(PXBINS+IPBIN+0)="::'
     IPRTBF(PXBINS+IPBIN+1)="::'
     IPRTBF(PXBINS+IPBIN+2)="::'
     IPRTBF(PXBINS+IPBIN+3)="::'
390  IPBIN=IPBIN+3
400 CONTINUE
C
C RETURN TO CALLING ROUTINE
C
900 RETURN
END
SUBROUTINE P1CLI 0 LIST GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)

CALL MDNOTE
CALL NVIATO

EXTERNAL PIC129,PICL19
CALL TRACE

CALL MDNOTE:
= 'LIST,GRADIENT/LAPLACIAN/VARIANCE NOT YET IMPLEMENTED'
CALL NVIATO(PIC129,PICL19)
RETURN
END
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/RUTINE

SUBROUTINE PICLIS & LIST CLASS (PHASE 5)

(C. H. SCHLOSSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

EXTERNAL PIC129,PIC119

CALL TRACE

CALL MONOTE('LIST,CLASS NOT YET IMPLEMENTED')
CALL NVIATO(PIC129,PIC119)
RETURN
END
SUBROUTINE PICLIS  & LIST MS8-DERIVED DATA (PHASE 9)

HISTORY
---
E M SCHLOSSER LEC 02/23/79 ORIGINAL CODE
E M SCHLOSSER LEC 09/24/79 UPGRADE DOCUMENTATION
E M SCHLOSSER LEC 09/01/79 DON'T INCREMENT GENERATION IF DIAGNOSTIC

METHOD
-----
CHECK DIAGNOSTIC COUNTERS AND PREPARE FOR NEXT LIST.

MACHINE-DEPENDENT CODE
----------------------
NONE.

EXTERNAL REFERENCES
-------------------
MONOTE
MDCLRW

EXCEPTIONS
----------
NONE.

GLOBAL DECLARATIONS
-------------------
INCLUDE KOMXQT.LIST  & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST  & DEFINE NULL CHARACTER STRING
EXTERNAL PIC000.NULSUB

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

ON RETURN, CALL PIC000 TO GET COMMANDS
CALL NVIATO(PIC000,NULSUB)

L-117
C ANY DIAGNOSTICS???
C
980 IF(INDFATL.EQ.0) GO TO 920
   CALL HNOTE( 'FATAL ERRORS -- NO LIST GENERATED')
   GO TO 990
920 IF(INDMNAR.EQ.0) GO TO 980
   CALL HNOTE( 'PREVIOUS WARNINGS -- NO LIST GENERATED')
   IF(INMBATCH.EQ.0) WRITE(6,925)
   FMTA('**TRY AGAIN**)')
   CALL MDCLAM('NULCST')
   GO TO 990
C
C PREPARE FOR NEXT WINDOW
C
980 NWINDOW=NWINDOW+1
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
990 RETURN
END

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE PICPAR: A PARTITION FACTOR SPACE (PHASE O)
U KOND) 0: I: FIRST 3 CHAR OF COMMAND 0: SPACES

C (E H SCHLOSSER)

C EXTERNAL SUBROUTINES/FUNCTIONS CALLED

READS
MONARN
CALSAA
CALWIN
CROPWR
OPPRPIC
RDUST
IDERT
IDRPIE
HDUNIT

INCLUDE KONKLLIST
INCLUDE KONKREWLIST
INCLUDE KONKLLIST
INCLUDE KONKLLIST
INCLUDE KONKLLIST
INCLUDE KONKLLIST
INCLUDE KONKLLIST
INCLUDE KONKLLIST
INCLUDE TRFORM.LIST
EXTERNAL PICT00.PIC129.PIC345.PICPA3.PICPA4.PICPA9.NULSUB
CALL TRACE

C GET/CHECK CRITERION & NAME NEXT 'TO' SUBROUTINE

IF(NNOWN.EQ.0) CALL MONARN('INVALID DEFAULT COMMAND')
KLISTP=0
CALL GETSKH(KLISTP,3,-0)
IF(KLISTP.NE.'DEN') GO TO 240
CALL NVIAT0(PIC345,PICPA3) 0 DENSITY -- NEXT CALL IS TO PICPA3
GO TO 300
240 IF(KLISTP.NE.'ORA') GO TO 250
CALL NVIAT0(PIC345,PICPA4) 0 GRADIENT -- NEXT CALL IS TO PICPA4
GO TO 300
250 IF(KLISTP.NE.'LAP') GO TO 280
CALL NVIAT0(PIC345,PICPA4) 0 LAPLACIAN -- NEXT CALL IS TO PICPA4
GO TO 300
280 CALL WARN5('BAD PARTITION CRITERION --')

C DRAIN SPECS FOR CURRENT COMMAND

L-119
300 CALL GETSINITEMP,'1,1,'EXTRA PARTITION SPECIFICATION --'

C CHECK RADIANCE LIMITS
C IF(LCVL01.GT.LCVH11) CALL MDWARN('NO RADIANCE LIMITS')
C IF(MDATA0C.NE.0) GO TO 850 8 DATA/CHECKOUT MODE

C CALIBRATE SPACING/WINDOW
C CALL CALSPA
C CALL CALWIN(0.)

C CROP OUTPUT WINDOW TO FIT INPUT WINDOW
C CALL CROPWIN(2500,3500)

C CHECK FOR DIAGNOSTICS
C IF(NDOTOTL.NE.0) GO TO 800

C CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C IF(NWINNOW.LT.0) CALL OPRPIC 2 OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
NWINNOW=ABS(NWINNOW)
NPAGE=0

C PRINT WINDOW HEADING FOR UNIT 8
C IF(NBATCH.NE.0) CALL MDUNIT(4,8)
WRITE(8,415) NWINNOW,AMATERAL
415 FORMAT('WINDOW NUMBER ','J3,8X,'PARTITION','8X,4A8')
CALL IDLUS(8)
CALL IODET(8)
CALL IDCPIC(8)

C PRINT WINDOW HEADING FOR UNIT 10
C 450 CALL MDUNIT(4,10)
WRITE(10,415) NWINNOW,AMATERAL
CALL IDLUS(10)
CALL IODET(10)
CALL IDCPIC(10)
GO TO 800

C WARNINGS ENCOUNTERED -- NEXT CALL IS TO PICPAR
C 800 CALL NNIAT0PIC(129,PICPAR)
KLSTYP=0 8 NO PARTITION GENERATED
GO TO 900

DATA/CHECKOUT MODE -- GET NEXT COMMAND

950 CALL NVIATO(PIC000. NULSUB)

RETURN FOR CALL TO NAMED SUBROUTINE

900 KOMD='''
    RETURN
    END
SUBROUTINE PICPA3  & PARTITION BY DENSITY (PHASE 3)

E H SCHLOSSER

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

MDNOTE
NVIAE

EXTERNAL PIC129,PICPA9
CALL TRACE

CALL MDNOTE( 'PARTITION,DENSITY NOT YET IMPLEMENTED')
CALL NVIAEO( PIC129.PICPA9)
RETURN
END
SUBROUTINE PICPA9 - PARTITION BY GRADIENT/LAPLACIAN/VARIANCE (PHASE 4)

(C. H. SCHLOSSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

MONOTE
NVIAITO

EXTERNAL PIC129,PICPA9
CALL TRACE

CALL MONOTE(
"PARTITION,GRADIENT/LAPLACIAN/VARIANCE NOT YET IMPLEMENTED")
CALL NVIAITO(PIC129,PICPA9)
RETURN
END
SUBROUTINE PICPAS 6 PARTITION FACTOR SPACE (PHASE 6)

(E H SCHLOSSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

-HNOTE
-NVIATO

EXTERNAL PIC129, PICPAS
CALL TRACE

CALL HNOTE('PARTITION COMMAND NOT YET IMPLEMENTED')
CALL NVIATO(PIC129, PICPAS)
RETURN
END
SUBROUTINE PICPAB & PARTITION FACTOR - PAGE (PHASE 0)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

MONOTE
NVIATO

EXTERNAL PICOOO.NULSUB
CALL TRACE

CALL MONOTE("PARTITION COMMAND NOT YET IMPLEMENTED")
CALL NVIATO(PICOOO.NULSUB)
RETURN
END
SUBROUTINE PICPIC \ 0 PICTURE RADIANCE/GRADE/ALPLACIAN/VARIANCE/CLASS
U KOND) \ 0 1: FIRST 3 CHAR OF COMMAND E: SPACES

HISTORY

E H SCHLOSSER LEC 09/17/79 REQUIREMENTS & DESIGN
E H SCHLOSSER LEC 11/06/79 STUBBED
E H SCHLOSSER LENSCO 09/20/80 IMPLEMENTED

METHOD

CHECK/CALIBRATE SPECs, CLEAR SCREEN, & GENERATE PICTURE HEADINGS.
THE NAME PICPIC/9 TO GENERATE BODY OF PICTURE.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETCH 0 GET CHARACTER STRING DATA FIELD FROM UNIT 9
GETSIN 0 GET INTEGER DATA FIELD FROM UNIT 9
HOMARN 0 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
HOMOTE 0 PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
INIATO 0 NAME 'VIA' 'TO' SUBROUTINES
CALCAL 0 CALIBRATE CHANNELS
CALCOL 0 CALIBRATE COLOR/INTENSITY SYMBOLS
CALSPA 0 CALIBRATE TRANSFORMATION COEFFICIENTS FOR SPACING
CALWIN 0 CALIBRATE WINDOW ENVELOPES
CROPW 0 CROP OUTPUT WINDOW
EXAPNT 0 WRITE ASCII IMAGE TO PRIMARY OUTPUT DEVICE
QPRPIC 0 OPEN ALTERNATE PRINT FILE(S)
RDSL 0 IDENTIFY LOGICAL UNIT 3
IDCMT 0 IDENTIFY CRTS SCENE
IDCPIC 0 IDENTIFY CURRENT COMMAND SPECS FOR PICTAB
HOUNIT 0 WRITE HEADING LINE(S) AT TOP OF NEXT PAGE
WARM 0 SUBMIT WARNING FOR MISSING/INVALID FIELD FROM UNIT 9

VIAT EXTERNAL PICPAGE, NULSUB
EXTERNAL PIC120, PICPI
EXTERNAL PIC140, PICPIC, PICPIC

EXCEPTIONS

1. "PICTURE" MAY NOT BE A DEFAULT COMMAND.
2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE PICTURE.

3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS

<table>
<thead>
<tr>
<th>CONC</th>
<th>ACTION</th>
<th>DIAGNOSTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESSING DEFAULT COMMANDS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(WINDOW=0)</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>KLISTYP IN COMMON UNEFIND</td>
<td>KLISTYP= 'RAD'</td>
<td></td>
</tr>
<tr>
<td>KLISTYP SPECIFICATION MISSING</td>
<td>USE COMMON KLISTYP</td>
<td></td>
</tr>
<tr>
<td>KLISTYP SPECIFICATION INVALID</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>EXTRA SPECIFICATION</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>LIMIT CHANNEL VALUE RANGE IS NULL</td>
<td>LCVL01&gt;LCVH11</td>
<td></td>
</tr>
<tr>
<td>COLOR MODE SWITCH NOT ON</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>BATCH RUN</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>DATA/CHECKOUT MODE</td>
<td>'TO' ROUTINE IS NULSUB</td>
<td>NONE</td>
</tr>
<tr>
<td>WARNING(S) OR FATAL ERROR(S)</td>
<td>'TO' ROUTINE IS PICPI9</td>
<td>NONE</td>
</tr>
</tbody>
</table>

GLOBAL DECLARATIONS

- INCLUDE KOMXGT.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 KOMKLS.LIST - COMMON COLOR SCREEN PARAMETERS
- INCLUDE KOMOWL.LIST - COMMON OUTPUT WINDOW PACKETS
- INCLUDE NULCST.LIST - DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

- INTEGER KLSAVE - SAVE LOCATION FOR KLISTYP
- INTEGER INTEMP - TEMPORARY
- INTEGER NPRLIN, NPRCOL - NUMBER OF PPD LINES/COLUMNS IN PICTURE

PROCEDURE

- CALL TRACE

CHECK IF COMMAND IS LEGAL

IF (WINDOW.EQ.0) CALL MDWARN ("INVALID DEFAULT COMMAND")
IF (BATCH.NE.0) CALL MDWARN ("PICTURE COMMAND NOT ALLOWED IN BATCH")
IF (INCOLOR.EQ.0) CALL MDWARN (""
**PICPIC**

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

C GET PICTURE TYPE

C KTBLTY='NUL' & MARK OLD FREQ TABLE AS DESTROYED
KLSAVE=KLSTyp & SAVE PREVIOUS PICTURE TYPE

IF(KLSTyp.NE.'ORA').AND.
& (KLSTyp.NE.'LAP').AND.
& (KLSTyp.NE.'VAR').AND.
& (KLSTyp.NE.'CLA'))
   8 IF UNDEFINED ...
   8 ... THEN MAKE IT RADIANCE ...
   CALL GETSKM(KLSTyp,(3), NULCST) & ... UNLESS SPECIFIED BY PICTURE CMD

C CHECK PICTURE TYPE

C IF(KLSTyp.NE.'RAD') GO TO 240 & RADIANCE?
   CALL NVIATOC PICS45,PICP13) & NEXT CALL IS TO PICP13
   GO TO 300

240 IF(KLSTyp.NE.'ORA').AND.
& (KLSTyp.NE.'LAP').AND.
& (KLSTyp.NE.'VAR').GO TO 250 & VARIANCE?
   CALL NVIATOC PICS45,PICP14) & NEXT CALL IS TO PICP14
   GO TO 300

250 IF(KLSTyp.NE.'CLA') GO TO 260 & CLASS?
   CALL NVIATOC PICS45,PICP15) & NEXT CALL IS TO PICP15
   GO TO 300

260 CALL WARNS('BAD PICTURE TYPE --')
   KLSTyp=KLSAVE & RESTORE PREVIOUS PICTURE TYPE

C DRAIN SPECS FOR CURRENT COMMAND

C 300 CALL GETSINI(INTEMP, +1.1.'EXTRA PICTURE SPECIFICATION --')

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 CALIBRAGE CHANNELS/COLORS/SPACING/WINDOW

C CALL CALCHA
C CALL CALCOL
C CALL CALSPA
C CALL CALWINI 0.1

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

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

L-128
COMMON PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

L-129

PAGE=0

C CROP OUTPUT WINDOW TO FIT INPUT WINDOW & COLOR SCREEN
CALL CROPWIN(KSLINE,KSCOLM)
IF (NDTOLNE.0) GO TO 900

C COMPUTE SIZE OF PRINT/PLT DEVICE (COLOR SCREEN) WINDOW
NPRLIN=IFIX(PPOWW(LWIN.WMAXI)-IFIX(PPOWW(LWIN.WMINI)+1
NPRCOL=IFIX(PPOWW(WCOL.WMAXI)-IFIX(PPOWW(WCOL.WMINI)+1

C CHECK FOR DIAGNOSTICS
IF (NDTOLNE.0) GO TO 900

C CLEAR SCREENS AND PRINT WINDOW HEADING
CALL EAPRNT(0.1.KSON) ROUTE UNIT 6 OUTPUT TO COLOR SCREEN
CALL EAPRNT(0.1.KSCLER) CLEAR COLOR SCREEN
WRITE(6,415) NWNDOW.MTEGR
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.MTEGR
CALL IDLU3(6)
CALL IDERTI(6)
CALL IDCPI(6)

C ANY DIAGNOSTICS???
900 IF (NDTOLNE.0) GO TO 990
IF (NDTACLNE.0) CALL NVIAT0(PIC000.NULSUB) DATA/CHECKOUT
IF (NDTACLNE.0) CALL NVIAT0(PIC129.PICPI9)

C RETURN FOR CALL TO NAMED SUBROUTINE
990 KOND="#
RETURN
END
SUBROUTINE PICP13 & PICTURE RADIANCE (PHASE 3)

HISTORY
------
E H SCHLOSSER LEC 05/17/79 REQUIREMENTS & DESIGN
E H SCHLOSSER LEC 11/06/79 STUBBED
E H SCHLOSSER LEMSCO OS/28/80 IMPLEMENTED

METHOD
-------
INITIALIZE LOW AND HIGH PPD LINES AND COLUMNS. INITIALIZE LOW AND
HIGH ADJUSTED LINE. CALL OETRAD TO READ LINE. MASK NON-TRIVIAL
WINDOW. RESAMPLE/SCREEN/COUNT FREQUENCY/SYMBOLIZE LINE. OUTPUT
LINE. NAME PICP19 AS 'TO' ROUTINE FOR WRAP-UP OF PICTURE
PROCESSING.

MACHINE-DEPENDENT CODE
----------------------
NONE.

EXTERNAL REFERENCES
-------------------
AVP ADJUSTED COORD FOR PRINT/PLLOT COORD
GETRAD GET ALL SELECTED RAW/TRANSFORMED CHANNELS
MSKPIX MASK NON-TRIVIAL WINDOW
NDPATL PRINT/LOG/COUNT 'FATAL ERROR' MESSAGES
NVIAO NAME 'VIA' 'TO' SUBROUTINES
EAPRNTO WRITE ASCII IMAGE TO PRIMARY OUTPUT DEVICE
EXTERNAL PIC129, PIC19
EXTERNAL RESCOL, GETBYT, GETICE, GETINT, GETNUL, ROUTINE TO GET BIN VALUE

EXCEPTIONS
---------
STATUS
FROM
GETRAD PPD CELL SYMBOLS DIAGNOSTIC ACTION
'EOL' NO DATA (GRAY) NONE WRITE LINE
'BADR' NO DATA (GRAY) NONE WRITE LINE
'BADF' N/A FATAL RETURN
'OFL' N/A FATAL RETURN

L-130
GLOBAL DECLARATIONS

INCLUDE KOMXQT.LIST  # COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST  # COMMON CLASSIFICATION INFO
INCLUDE KOMSYN.LIST  # COMMON SYMBOL TABLE
INCLUDE KOMKS.LIST   # COMMON COLOR SCREEN PARAMETERS
INCLUDE WINDEF.LIST  # DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMOWN.LIST  # COMMON OUTPUT WINDOW PACKETS
INCLUDE KORTBL.LIST  # COMMON TICK/FREQ TABLE
INCLUDE PICDEF.LIST  # DEFINE PICTAB PARAMETERS
INCLUDE PXBDEF.LIST  # DEFINE PIXEL BUFFER STRUCTURE
INCLUDE MAXINT.LIST  # MAXIMUM INTEGER VALUE

LOCAL DECLARATIONS

PARAMETER NXBUFS=6  # OF MSS PIXEL BUFFERS IN ARRAY
INTEGER IN MSA BUF = #INTS PREAMBLE * (#INS+3)/4 * (#EXTRA BYTES+3)/4
PARAMETER NWIXBF  = (#XINS-1) * (#EXTRA BYTES+3)/4 * (#INS+3)/4
PARAMETER NHIKBF = (#XINS-1) * 256
PARAMETER NFRPCH=8  # WDS PER CHANNEL IN LOCAL FREQ TABLE
PARAMETER NFRQSZ=128 # CHANNELS IN LOCAL FREQ TABLE
INTEGER MPXBUFINWIXBF.NXBUFS)  # ARRAY OF MSS PIXEL BUFFERS
INTEGER KIBUF(NWIKBF)  # COLOR/INTENSITY BUFFER
INTEGER NFRPCH(NFRQSZ,NFRPCH)  # LOCAL FREQ TABLE (SCOPE INCLUDES
INTEGER NFRCRO(NFRQSZ.NFRPCH)  # INTERNAL ROUTINE RESCOL)
INTEGER IN IFRQRO(10,15)  # LOCAL CROSS FREQ TABLE
INTEGER IPLIN  # PPD LINE
INTEGER IPCMIN.IPCMAX # MINIMUM AND MAXIMUM PPD COLUMN
INTEGER IPLMIN.IPLMAX # MINIMUM AND MAXIMUM PPD LINE
REAL ADJLIN,ADJSAM  # ADJUSTED LINE AND SAMPLE
INTEGER MSLIN.MLHSH.MLHSS  # MSA LINE*100: LOW,HIGH,SPACING
INTEGER MSLIN  # 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 IRSLIN  # LAST SCAN LINE READ

PROCEDURE

CALL TRACE

DECLARE MINIMUM AND MAXIMUM PPD LINES AND COLUMNS

IPLMIN=PPDOWN(MIN,WMIN,WMAX)
IPLMIN=PPDUP(MIN,WMAX,WMIN)
IPCHIN=PPDUP(WCOL,WHIN)
IPCHIN=PPDUP(WCOL,WHIN)
C COMPILE NUMBER OF PPD LINES AND COLUMNS
C
NPRLIN=IPLMAX-IPLMIN+1
NPRCOL=IPCHMAX-IPCHMIN+1
C CLEAR LOCAL FREQUENCY TABLES
C
DO 250 I=1,NLIMCH
DO 250  K=1,NFRQSZ
  NFREQ(K,1)=0
250 CONTINUE
DO 250  I=1,NL100
K=1,MAXINT
  NFRCRO(K,1)=0
270 CONTINUE
C INITIALIZE LOW AND HIGH LINES AND SPACING
C
CALL A4P (ADJLIN,ADJSAM, ML100,ADJLIN+100.
CALL A4P (ADJLIN,ADJSAM, ML100,ADJLIN+100.
ML100=MSADWIN(WLIN,WSP100)
C POSITI ON AT TOP OF WINDOW
C
MSALIN=ML100/100
CALL GETRAD (MPKBUF,(0),(NXBUFS),ISTAT, MSALIN.0.0)
IF (ISTAT.NE. ''BADF'' ) GO TO 300
  CALL HDEFATL ( ''BADF (BAD FILE) ON UNIT 3'')
GO TO 900
300 CONTINUE
C TURN ON COLOR SCREEN
C
CALL EAPRNT(O,1,KSON)
C READ, MASK, RESAMPLE, SCREEN, SYMBOLIZE AND WRITE SCAN LINES
C
IP LIN=IPLMIN
LASTLN=MAXINT
DO 380 ML100=ML100,ML100,M100S
MSALIN=ML100/100
CALL A4P (ADJLIN,ADJSAM, ML100,ADJLIN+100.
MSASLO=ADJSAM
CALL A4P (ADJLIN,ADJSAM, ML100,ADJLIN+100.
MSASN=ADJSAM
IF (MSALIN.NE.LASTLN) CALL GETRAD (MPKBUF,(NWXBF),(NXBUFS).
LASTLN = MSALIN
IF ( (ISTAT.NE. 'BADF') .AND. (ISTAT.NE. 'OFI')) GO TO 320
   CALL HOFATL (COS4CS(ISTAT.1.4)
   * WHILE READING ON UNIT 3*
   GO TO 900
320 CALL MSKPIX (MPXBUF(1,1), MPXBUF(1,1))
   IF ( (MPXBUF(PIXINT,1).EQ. 'BYT').OR.
   (MPXBUF(PIXINT,1).EQ. 'CHR').OR.
   (MPXBUF(PIXINT,1).EQ. 'INT').OR.
   (MPXBUF(PIXINT,1).EQ. 'NUL')) GO TO 330
   CALL HOFATL('INVALID BIN TYPE',
   COS4CS(MPXBUF(PIXINT,1).1.3). IN PICP13')
   GO TO 900
330 IF (MPXBUF(PIXINT,1).EQ. 'BYT') CALL RESCOL (KIBUF,(1PLIN),
   (1PCMIN), (1PCMAX), MPXBUF,(NMXBIF),(NXMLFS),GETBYT)
   IF (MPXBUF(PIXINT,1).EQ. 'CHR') CALL RESCOL (KIBUF,(1PLIN),
   (1PCMIN), (1PCMAX), MPXBUF,(NMXBIF),(NXMLFS),GETICE)
   IF (MPXBUF(PIXINT,1).EQ. 'INT') CALL RESCOL (KIBUF,(1PLIN),
   (1PCMIN), (1PCMAX), MPXBUF,(NMXBIF),(NXMLFS),GETINT)
   IF (MPXBUF(PIXINT,1).EQ. 'NUL') CALL RESCOL (KIBUF,(1PLIN),
   (1PCMIN), (1PCMAX), MPXBUF,(NMXBIF),(NXMLFS),GETNUL)
   CALL EAPRNT(1,N14NB(KIBUF,PIXBIN1),KIBUF,PIXBIN)
   1PLIN = 1PLIN + 1
350 CONTINUE
C C C MOVE DATA FROM LOCAL FREQ TABLES TO COMMON TABLE (REPLACING TICKS)
C KTBLTY = 'FREQ'
   KTBLNM = 'NFWOON'
   DO 600 I=1,NL1RCM
      DO 590 K=1,NFRQSZ
         KFREQ(K,1) = NFREQ(K,1)
   590 CONTINUE
   600 CONTINUE
   DO 700 I=1,110
      DO 690 K=1,15
         KFRCRO(I,K) = NFRCRO(I,K)
   690 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 MAVIATO ( PIC129, PICP19)
   RETURN

L-133
INTERNAL

SUBROUTINE RESCOL: 8 RESAMPLE/SCREEN/COUNT FREQUENCY/COLOR
(a) KIBUF: 8 COLOR/INTENSITY BUFFER
(b) IPLIN: 8 PPD LINE
(c) IPCHIN: 8 MINIMUM PPD COLUMN
(d) IPCHMAX: 8 MAXIMUM PPD COLUMN
- 1 MPXBUF: 8 ARRAY OF MSS PIXEL BUFFERS
- 1 NHIXBF: 8 NUMBER OF WORDS IN ONE PIXEL BUFFER
- 1 NLSBF: 8 NUMBER OF PIXEL BUFFERS
- 1 GETBIN: 8 ROUTINE TO GET BIN VALUE--GETBYT, GETICE, GETINT, GETNUL

METHOD

COMPUTE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER PREAMBLE.
FOR EACH SAMPLE, CHECK IF OUTSIDE OF INPUT WINDOW. CHECK IF OUTS
SIDE OF RADIANCE LIMITS. COUNT FREQUENCY. AND SYMBOLIZE. INSERT

EXTERNAL REFERENCES

AMP 8 ADJUSTED MSS COORD FOR PRINT/PLOT COORD
GETICE 8 GET INTEGER-CHARACTER-EQUIVALENT FROM CHARACTER STRING
PUTBYT 8 PUT BYTE INTO BYTE STRING
MOVBST 8 MOVE BYTE STRING

GLOBAL DECLARATIONS

#include KOMOW.LIST 8 COMMON OUTPUT WINDOW PACKETS
#include KOMTB.LIST 8 COMMON FREQ/TICK TABLE
#include KOMKLS.LIST 8 COMMON CLASSIFICATION INFO
#include PXBDEF.LIST 8 DEFINE PIXEL BUFFER STRUCTURE
#include KOMSYM.LIST 8 COMMON SYMBOL TABLE
#include KOMKLS.LIST 8 COMMON COLOR SCREEN PARAMETERS
#include WINDEF.LIST 8 DEFINE WINDOW PACKETS
#include MAXINT.LIST 8 DEFINE MAXIMUM INTEGER

LOCAL DECLARATIONS

PARAMETER NUMBFS=6 8 NUMBER OF MPX BUFFERS
INTEGER MPXBUF(NHIXBF,NLSBF) 8 ARGUMENT
INTEGER KIBUF(1) 8 ARGUMENT
INTEGER NHINS(NUMBFS) 8 BIN NUMBER OF SAMPL FOR EACH MPX BUFFER
INTEGER MS100L,MS100M,MS100S 8 MSA SAMPLE*100; LOW,HIGH,SPACING
INTEGER MSASAM 8 MSA SAMPLE NUMBER
INTEGER KIBIN 8 COLOR/INTENSITY BUFFER BIN POINTER
REAL ADJSAM 8 ADJUSTED SAMPLE NUMBER
INTEGER IPXIL1,IPXIL2,IPXIL3,IPXIL4,IPXIL5,IPXIL6 8 PIXEL VALUE FOR EACH BUFFER
INTEGER IKENU 8 INTEGER-COLOR-EQUIVALENT (NEW)
INTEGER KIKE, KIKENU 8 CODED INTEGER-COLOR-EQUIVALENT (CURRENT), (NEW)
INTEGER IIE, IIEU
INTEGER KIIE
INTEGER INTENSITY-EQUIVALENT (CURRENT, NEW)
INTEGER KIIE
INTEGER CODED INTEGER-INTENSITY-EQUIVALENT (CURRENT)

C C PROCEDURE
C C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C CALL ASY (AOJLIN, AOJSAM, FLOAT(IPLIN), FLOAT(IPCHNI))
MS100L = AOJSAM - 100.
CALL ASY (AOJLIN, AOJSAM, FLOAT(IPLIN), FLOAT(IPMAX))
MS100N = AOJSAM + 100.
MS100S = MSASAM (MSAM, MS100)

C C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C DO 150 NUMBUF = 1, MXBUFF
	nsbins (numbuf) = mpbuf (potlin, numbuf) - mpbuf (potlin, numbuf)

C C INITIALIZE PREAMBLE FOR COLOR-INTENSITY BUFFER
C KIBUF (PXRCHN) = MPXBUF (PXRCHN, 1)
KIBUF (PXLIN) = IPLIN
KIBUF (PXCHAN1) = 0
KIBUF (PXQUAL) = 0
KIBUF (PXBIN) = 'BYT'
KIBUF (PLBIN) = 1
KIBUF (PXLIN) = IPLIN
KIBUF (PXHOL) = IPCOL
KIBUF (PXR01N) = 0
KIBUF (PX0D0) = 0
KIBUF (PXL01) = 0
KIBUF (PXNL0) = 0

C C INITIALIZE BIN POINTER & FIRST BIN OF COLOR-INTENSITY BUFFER
C KIBIN = KIBUF (PXLBIN)
CALL PUTBYT (KIBUF (PXRCHN1) (KIBIN), 33) & 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
KIIE = MAXINT

C C RESAMPLE/SCREEN RADIANCE/COUNT FREQUENCY/LOOK UP SYMBOLS
C DO 462 MS100 = MS100L, MS100H, MS100S
MSASAM = MS100 / 100

L-135
C BUFFER 1

C
IF (I(MSASAM.LT.MPXBUF(PXLSAM.1))). OR. 
   (I(MSASAM.GT.MPXBUF(PXNSAM.1))) GO TO 390 8 SAMPLE NOT IN BUFFER
   CALL GETIN (IPIXL1.
   *  MPXBUF(PXBIN(1). (MSASAM=NBINSO(1))).
IF (IPIXL2=IPIXL1) 8 SECOND CHANNEL VALUE SAME AS FIRST IF UNDEFINED
IF (IPIXL1.GE.MPXBUF(PXMODA.1))) GO TO 350 8 NO DATA
   IF (IPIXL1.LT.LCVL011). OR.
   (IPIXL1.GT.LCVH111)) GO TO 360 8 OUT OF RAD LIMITS
   IF (NLINCH-1.EQ.0) GO TO 320

C BUFFER 2

C
IF (I(MSASAM.LT.MPXBUF(PXLSAM.2))). OR. 
   (I(MSASAM.GT.MPXBUF(PXNSAM.2))) GO TO 350 8 SAMPLE NOT IN BUFFER
   CALL GETIN (IPIXL2.
   *  MPXBUF(PXBIN(2). (MSASAM=NBINSO(2))).
IF (IPIXL2.LT.LCVL022). OR.
   (IPIXL2.GT.LCVH222)) GO TO 360 8 OUT OF RAD LIMITS
   IF (NLINCH-2.EQ.0) GO TO 290

C BUFFER 3

C
IF (I(MSASAM.LT.MPXBUF(PXLSAM.3))). OR. 
   (I(MSASAM.GT.MPXBUF(PXNSAM.3))) GO TO 350 8 SAMPLE NOT IN BUFFER
   CALL GETIN (IPIXL3.
   *  MPXBUF(PXBIN(3). (MSASAM=NBINSO(3))).
IF (IPIXL3.LT.LCVL033). OR.
   (IPIXL3.GT.LCVH333)) GO TO 360 8 OUT OF RAD LIMITS
   IF (NLINCH-3.EQ.0) GO TO 260

C BUFFER 4

C
IF (I(MSASAM.LT.MPXBUF(PXLSAM.4))). OR. 
   (I(MSASAM.GT.MPXBUF(PXNSAM.4))) GO TO 350 8 SAMPLE NOT IN BUFFER
   CALL GETIN (IPIXL4.
   *  MPXBUF(PXBIN(4). (MSASAM=NBINSO(4))).
IF (IPIXL4.LT.LCVL044). OR.
   (IPIXL4.GT.LCVH444)) GO TO 360 8 OUT OF RAD LIMITS
   IF (NLINCH-4.EQ.0) GO TO 230

C BUFFER 5

C
IF (I(MSASAM.LT.MPXBUF(PXLSAM.5))). OR. 
   (I(MSASAM.GT.MPXBUF(PXNSAM.5))) GO TO 350 8 SAMPLE NOT IN BUFFER
   CALL GETIN (IPIXL5.
   *  MPXBUF(PXBIN(5). (MSASAM=NBINSO(5))).
IF (IPIXL5.LT.LCVL055). OR.
   (IPIXL5.GT.LCVH555)) GO TO 360 8 OUT OF RAD LIMITS
   IF (NLINCH-5.EQ.0) GO TO 200

C BUFFER 6

C
IF (I(MSASAM.LT.MPXBUF(PXLSAM.6))). OR.
CAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C (RUNSAM.OT.HPKBUN(PKHSAM.6) GO TO 350  S SAMPLE NOT IN BUFFER
    CALL GETBIN (IPXL6).
    HPKBUN(PKBSNS.6),(RUNSAM-NBINSO(6))
    IF ((IPXL6.LT.LCVLO(6)).OR.
       (IPXL6.LT.LCVHI(6))) GO TO 380  O OUT OF RAD LIMITS

C COUNT FREQUENCY
    NFRQ(IPXL6+1.0)=NFRQ(IPXL6+1.0)+1
    NFRQ(IPXL5+1.5)=NFRQ(IPXL5+1.5)+1
    NFRQ(IPXL4+1.4)=NFRQ(IPXL4+1.4)+1
    NFRQ(IPXL3+1.3)=NFRQ(IPXL3+1.3)+1
    NFRQ(IPXL2+1.2)=NFRQ(IPXL2+1.2)+1
    NFRQ(IPXL1+1.1)=NFRQ(IPXL1+1.1)+1

C INFO -- LOOK UP NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
    CALL GETIC:E:11ENU.
    KSYNI(IPXL1+1.5)=(5)
    CALL GETIC:E:1KENU.
    KSYM(I(IPXL2+1.1)=(6)
    GO TO 370

C NO DATA -- ASSIGN NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
    CALL GETIC:E:11ENU.
    KSYNI(I(ISYMNO=1.5)=(5)
    CALL GETIC:E:1KENU.
    KSYM(I(ISYMNO=1.1)=(6)
    GO TO 370

C NO INFO -- ASSIGN NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV
    CALL GETIC:E:11ENU.
    KSYM(I(ISYMNI=1)=(5)
    CALL GETIC:E:1KENU.
    KSYM(I(ISYMNI=1)=(6)

C COUNT INTENSITY X COLOR CROSS FREQUENCY
    NFRCRO(I(IENU=1,1KENU=1)=NFRCRO(I(IENU=1,1KENU=1)+1

C IF NEW CODED 1-K-E, PUT CODED 1-K-E & CODED 1-1-E IN BUFFER
    KIKENU=KSKEK(IKENU=1)
    IF(KIENU.EQ.KIKE) GO TO 380
    KIKE=IKENU
    KIBIN=KIBIN+1
    CALL PUTBYT(KIOUTIP KBINS).KIBIN. KIKE)
    IIE=IIENU
    KIE=KIE+1

L-137
KIBIN=KIBIN+1
CALL PUTBY(JKIBUF(PXBIN), KIBIN). KIE
GO TO 400

C
C SAME COLOR -- IF SAME INTENSITY & BIN, UPDATE CODED IIE IN BUFFER
C
380 IF(11E=1E .NE. 1IE) GO TO 380
1/(K1E .GT. 11E) GO TO 380
KIE=KIE-10
CALL PUTBY(JKIBUF(PXBIN), KIBIN). KIE
GO TO 400

C
C SAME COLOR WITH NEW INTENSITY OR NEW BIN
C
390 11E=11ENU
K1E=K1E+97
KIBIN=KIBIN+1
CALL PUTBY(JKIBUF(PXBIN), KIBIN). KIE

C
C LOOP TO PROCESS NEXT PIXEL
C
400 CONTINUE

C
C STORE POINTER TO LAST BIN IN PREAMBLE & PAD BUFFER WITH 8 NUL'S
C
KIBUF(PXBIN)=KIBIN
CALL MOVST(JKIBUF(PXBIN), KIBIN+11). (8).
.0.1(11),(11),0

C
500 RETURN

END
SUBROUTINE PICP19  & PICTURE GRADIENT/LAPLACIAN/VARIANCE (PHASE 9)

C (E H SCHLOSSER)

C EXTERNAL SUBROUTINES/FUNCTIONS CALLED

C

C C C C C

C

C

C C C C C

C

C

C

C C C C C

C

C

C

C C C C C

C

C

C

CALL NDNOTE

CALL NVIATO

EXTERNAL PIC120,PICP19

CALL TRACE

CALL NDNOTE!

"PICTURE,GRADIENT/LAPLACIAN/VARIANCE NOT YET IMPLEMENTED"

CALL NVIATO!  PIC120,PICP19

RETURN

END
SUBROUTINE PICPI8 A PICTURE CLASS (PHASE 5)

(E H SCHLOSSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

NONOTE

EXTERNAL PIC18D,PICPI8

CALL TRACE

CALL NONOTE( 'PICTURE,CLASS NOT YET IMPLEMENTED'

CALL NVIAT0 PIC18D,PICPI8)

RETURN

END
SUBROUTINE PICPI9 & PICTURE SCAN DATA (Phase 9)

HISTORY

E H SCHLOSSER  LEC  05/17/79  REQUIREMENTS & DESIGN
E H SCHLOSSER  LEC  11/06/79  STUBBED
E H SCHLOSSER  LEMSCO  05/28/80  IMPLEMENTED

METHOD

CHECK DIAGNOSTIC COUNTERS AND PREPARE FOR NEXT PICTURE.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

MONOTE
MOCLRW

EXCEPTIONS

NONE.

GLOBAL DECLARATIONS

INCLUDE KOMXOT.LIST  COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMONW.LIST  COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDFLLIST  DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE NULLST.LIST  DEFINE NULL CHARACTER STRING

CALL PIC000. NULSUB

PROCEDURE

CALL TRACE

ON RETURN, CALL PIC000 TO GET COMMANDS
CALL NVIATO(PIC000,NULSUB)

C
C ANY DIAGNOSTICS???
C
IF(NOFATL.EQ.0) GO TO 920
    CALL NOERROR('FATAL ERRORS -- NO PICTURE GENERATED')
    GO TO 990
920 IF(NOMAWR.EQ.0) GO TO 960
    CALL NOERROR('PREVIOUS WARNINGS -- NO PICTURE GENERATED')
    IF(MATCH.EQ.0) WRITE(6,925)
925 FORMAT(1X,'***TRY AGAIN***')
    CALL NOCLR(NULCST)
    GO TO 990
C
C PRINT MSA OUTPUT WINDOW COORDINATES AND PREPARE FOR NEXT WINDOW
C
960 WRITE(6,965) MSAOWW(WLIN,WMIN)
965 FORMAT(1X,1X,'LINE ',1X)
    WRITE(6,975) MSAOWW(WSAM,WMIN),MSAOWW(WSAM,WMAX)
975 FORMAT(1X,'SAMPLE ',14,9X,'SAMPLE ',14)
    WRITE(6,985) MSAOWW(WLIN,WMAX)
    WRITE(6,985)
985 FORMAT(1X)
    NNEW=NWINDOW+1
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
990 RETURN
END
SUBROUTINE PICPRO: 0 GENERATE PROFILES
U KOND) 0 1: FIRST 3 CHAR OF COMMAND 0 2: SPACES

C (E N SCHLOSSER)
C

INCLUDE KOMQT.LIST
EXTERNAL PIC000.NULSUB
CALL TRACE

CALL NVIATO(PICO00.NULSUB)
CALL MNOTE('PROFILE COMMAND NOT YET IMPLEMENTED')
RETURN
END
SUBROUTINE PICPR3 & PROFILE MSS-DERIVED DATA (PHASE 3)

C (E. M. SCHLOSSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

INCLUDING KOMTBL.LIST
EXTERNAL PIC129,PICPR9
CALL TRACE

CALL MDNOTE('PROFILE.RADIANCE NOT YET IMPLEMENTED')
CALL NVIATO(PIC129,PICPR9)
RETURN
END
SUBROUTINE PICPRO

SUBROUTINE PICPRO & PROFILE MSS-DERIVED DATA (PHASE 0)

(E. H. SCHLOSSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

EXTERNAL PIC000.NULSUB
CALL TRACE

CALL NDJNOTE('PROFILE NOT YET IMPLEMENTED')
CALL NIJATO(PIC000.NULSUB)
RETURN
END
SUBROUTINE PICROT: 6 ROTATE PREVIOUSLY GENERATED FACTOR STRUCTURE/COEFFS
U KOND1: 6 I: FIRST 3 CHAR OF COMMAND 0: SPACES

HISTORY
-------
M L BROWN  LEC 01/16/70  ALGORITHM CODING
J C CRISP  LEC 08/31/79  REVISE CALLS TO GETSSX

METHOD
-------
CHECK IF RESULTS OF FACTORING ARE AVAILABLE. GET/CHECK SPECS.
PERFORM ROTATION. PRINT ANGLE. ROTATED STRUCTURE. COEFFICIENTS.
MEANS.

MACHINE-DEPENDENT CODE
------------------------
NONE.

EXTERNAL REFERENCES
--------------------
MONOTE  a PRINT/LOG/COUNT 'NOTE' DIAGNOSTIC MESSAGES
MDWARN  a PRINT/LOG/COUNT 'WARNING' DIAGNOSTIC MESSAGES
MDCLRW  a CLEAR COUNT OF 'WARNING' DIAGNOSTICS
WARNS  a GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
GETSKH  a GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN  a GET INTEGER DATA FIELD FROM UNIT 5
GETSSX  a GET SEXADENARY DATA FIELD FROM UNIT 5
ROTCHX  a ROTATE 2 MATRIX COLUMNS TO MAXIMIZE FUNCTION
ROTCOL  a ROTATE 2 MATRIX COLUMNS
FACPRF  a PRINT FACTOR STRUCTURE/COEFFICIENTS/MEANS
EXTERNAL QUARTU, QUARTN, VARSQU, VARSQ

EXCEPTIONS
-----------
1. IF FACTOR TABLES ARE NOT PRESENT FROM A PREVIOUS FACTOR COMMAND,
   THEN A WARNING DIAGNOSTIC WILL BE GENERATED.
2. INVALID AND/OR EXTRA SPECIFICATION GENERATES A WARNING DIAGNOSTIC.
3. PREVIOUS WARNINGS AND/OR FATAL ERRORS PREVENT ROTATION.

GLOBAL DECLARATIONS
---------------------
INCLUDE KOKKOT.LIST  
INCLUDE KOKKONER.LIST  
INCLUDE KOKKLK.LIST  
INCLUDE KOKKBL.LIST  
INCLUDE KOKKIRL.LIST  
INCLUDE NULCST.LIST

LOCAL DECLARATIONS
---------------------
REAL ARAD  # ROTATION ANGLE IN RADIANS
REAL ADEG  # ROTATION ANGLE IN DEGREES

PROCEDURE
---------

CALL TRACE

CHECK IF FACTOR TABLE LOADED

IF((KLSTYP.EQ.0).OR.  
   (KSTYP.EQ. 'FACT') .AND.  
   (KTBLTY.NE. 'FACT'))  
   CALL MOWARN('NO PREVIOUS FACTORING')

GET FUNCTION TO MAXIMIZE UNDER ROTATION

KFUTYP='VAR'  # DEFAULT IS VARIMAX ROTATION
CALL GETSKH(KFUTYP,NULCST)
IF(KFUTYP.EQ.'DEG')  
   GO TO 500  # NO MAXIMIZATION -- EXPLICIT ANGLE
IF(KFUTYP.EQ. 'VAR'.AND.KFUTYP.EQ.?QUA')  
   CALL WARN('BAD MAXIMIZATION FUNCTION --')

GET TYPE OF MATRIX USED IN EVALUATING FUNCTION

KMATYP='STR'  # DEFAULT IS FACTOR STRUCTURE
CALL GETSKM(KMATYP,NULCST)
IF(KMATYP.EQ. 'STR'.AND.KMATYP.EQ.?COE')  
   CALL WARN('BAD EVALUATION MATRIX --')

IS NORMALIZED MATRIX TO BE USED?

KNOTYP='NOR'  # DEFAULT IS NORMALIZED
CALL GETSKH(KNOTYP,NULCST)
IF(KNOTYP.EQ. 'NOR'.AND.KNOTYP.EQ.?UNN'.AND.KNOTYP.EQ.?RAW')  
   CALL WARN('BAD NORMALIZATION SPEC --')

DRAIN SPECS FOR CURRENT COMMAND
CALL GETSIN(ITEMP,-1,-1,'EXTRA ROTATE SPECIFICATION --')

CHECK FOR DIAGNOSTICS

IF(MOATAC.NE.00 TO 900 DATA/CHECKOUT MODE
IF(NOTOTL.NE.00 TO 800

BRANCH ON FUNCTION SPECIFICATION

IF(KFUTYP.NE.'QUA') GO TO 300 VARIMAX

QUARTIMAX ROTATION

IF(KMATYP.NE.'STR') GO TO 240 COEFFICIENTS

QUARTIMAX ROTATION OF FACTOR STRUCTURE

IF(KNOTYP.NE.'NOR') GO TO 220 UNNORMALIZED

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

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

QUARTIMAX ROTATION OF FACTOR COEFFICIENTS

IF(KNOTYP.NE.'NOR') GO TO 260 UNNORMALIZED

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

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

L-148
C
C VARIMAX ROTATION
C
300 IF(KMATYP NE 'STR') GO TO 340 8 COEFFICIENTS
C
C VARIAX ROTATION OF FACTOR STRUCTURE
C
IF(KMATYP NE 'HOR') GO TO 320 8 UNNORMALIZED
C
WRITE(6,319)
319 FORMAT(' ROTATE. VARIAX. STRUCTURE. NORMALIZED'/)
CALL ROTCMX(FSTRUC,VARSON,KCHAC0,KFA000,7.7,1,2,ARAD,FSTROT)
CALL ROTCOL(FCNORM,KCHAC0,KFA000,7.7,1,2,ARAD,FCHNRT)
GO TO 700
C
C WRITE(6,329)
329 FORMAT(' ROTATE. VARIAX. STRUCTURE. UNNORMALIZED'/)
CALL ROTCMX(FSTRUC,VARSON,KCHAC0,KFA000,7.7,1,2,ARAD,FSTROT)
CALL ROTCOL(FCNORM,KCHAC0,KFA000,7.7,1,2,ARAD,FCHNRT)
GO TO 700
C
C VARIAX ROTATION OF FACTOR COEFFICIENTS
C
340 IF(KMATYP NE 'HOR') GO TO 360 8 UNNORMALIZED
C
WRITE(6,349)
349 FORMAT(' ROTATE. VARIAX. COEFFICIENTS. NORMALIZED'/)
CALL ROTCMX(FCNORM,VARSON,KCHAC0,KFA000,7.7,1,2,ARAD,FCHNRT)
CALL ROTCOL(FSTRUC,KCHAC0,KFA000,7.7,1,2,ARAD,FSTROT)
GO TO 700
C
C WRITE(6,369)
369 FORMAT(' ROTATE. VARIAX. COEFFICIENTS. UNNORMALIZED'/)
CALL ROTCMX(FCNORM,VARSON,KCHAC0,KFA000,7.7,1,2,ARAD,FCHNRT)
CALL ROTCOL(FSTRUC,KCHAC0,KFA000,7.7,1,2,ARAD,FSTROT)
GO TO 700
C
C GET EXPLICIT ROTATION ANGLE(S)
C
500 CALL GETSSX(ADEOFM,1..360..+360.. 'BAD ROTATE ANGLE ---')
C
C GET OPTIONAL FINAL ROTATION ANGLE & INCREMENT
C
ADEOT0=ADEOFM 8 DEFAULT FINAL ANGLE IS SAME AS INITIAL
CALL GETSSX(ADEOT0,1..360..+360.. 'BAD ROTATE FINAL ANGLE ---'
ADEOIN=1. 8 DEFAULT INCREMENT IS 1 DEGREE
CALL GETSSX(ADEOIN,1..360..+360.. 'BAD ROTATE INCREMENT ---')
IF (NDOTOTLE ' .NE. 0) GO TO 800  // DIAGNOSTICS ENCOUNTERED
WRITE (6, 915)

915 FORMAT (1X, 'PERFORM EXPLICIT ROTATION(s) & EVALUATE FUNCTIONS', /)
& '0 DEGREES   Q U A R T I M A X   V A R I M A X'/'
& 'STRUCTURE COEFFICIENTS STRUCTURE COEFFICIENTS'/'
& 'NORM UNNORM NORM UNNORM NORM UNNORM NORM UNNORM')
NDEOFM=ADEOFM+1000.
NDEGTO=ADEGTO+1000.
NDEGIN=ADEGIN+1000.
DO 540 NDEG=ADEOFM-NDEGTO,NDEGIN
ADEG=FLOAT(MDEG+0.001)
ARAD=ARAD+3.14159265/180.
CALL ROTCOL (FSTROCF, KCHAOC, KFAACC, 7, 7, 1, 2, ARAD, FSTROT)
CALL ROTCOL (FCNORM, KCHAOC, KFAACC, 7, 7, 1, 2, ARAD, FCNROT)
SUAR=SUAR(7, FSTROT, KCHAOC, KFAACC, 7, 7, 1, 2)
SUAR=SUAR(7, FCNROT, KCHAOC, KFAACC, 7, 7, 1, 2)
CUAR=CUAR(7, FSTROT, KCHAOC, KFAACC, 7, 7, 1, 2)
CUAR=CUAR(7, FCNROT, KCHAOC, KFAACC, 7, 7, 1, 2)
CSAR=CSAR(VARSON(7, FSTROT, KCHAOC, KFAACC, 7, 7, 1, 2)
CSAR=CSAR(VARSON(7, FCNROT, KCHAOC, KFAACC, 7, 7, 1, 2)
WRITE (6, 925) ADEG
& SUAR,SUAR,CUAR,CUAR,CUAR,CUAR,CURSON,CURSON
925 FORMAT (3, 1X, 'Y', 7.4, 1X, 'Y', 7.4, 1X, 'Y', 7.4, 1X, 'Y', 7.4, 1X, 'Y', 7.4)
540 CONTINUE
WRITE (6, 955)
955 FORMAT (1X)
GO TO 790
C
C PRINT ROTATION ANGLE IN DEGREES
C 790 ADEG=ARAD+3.14159265/180.
WRITE (6, 725) ADEG
725 FORMAT (1X, ' (ROTATION ANGLE = ', 9.4, ' DEGREES)' )
C
C PRINT FACTOR STRUCTURE/COEFFICIENTS/MEANS
C 790 CALL FACPRSTOT,FSTROT,FCOROT,CHSTD,CHMEAN,KCHAOC,KFAACC,7,7)
C
C ASSIGN FACTOR COEFFICIENTS TO LINEAR TRANSF WEIGHTS & UPDATE WEIGHTED GAINS
C DO 770 NCH=1, NERCHA
DO 780 NLF=1, 2
RTLWGT(NCH,NLF)=FCOROT(NCH,NLF)
770 LRTWGT(NCH,NLF)=RTLWGT(NCH,NLF)+RTLGAM(NLF)+2.*12
770 CONTINUE
C
C CHECK DIAGNOSTIC COUNTERS
C
800 IF(NDWMN.EQ.0) GO TO 820
   CALL MONE("PREVIOUS WARNINGS -- NO ROTATION")
   IF(NBATCH.EQ.0) WRITE(6,15)
   15 FORMAT(1S0,1S15) GO TO 900
820 IF(NDFATL.EQ.0) GO TO 850
   CALL MONE("PREVIOUS FATAL ERRORS -- NO ROTATION")
   GO TO 900
850 IF(NICHECK.EQ.0) GO TO 890
   CALL MONE("CHECKOUT MODE -- NO ROTATION")
C
C CLEAR WARNINGS
C
890 CALL MDCLRN(HULCST)
C
C RETURN
C
900 KOMO" RETURN
END
SUBROUTINE PICTOT( 8 TOTAL TABULATIONS
U KOND) 8 I: FIRST 3 CHAR OF COMMAND 0: SPACES

(E M SCHLOSSER)

INCLUDE KMKT.LIST
EXTERNAL PICOOD,HULSUB
CALL TRACE

CALL HVIAIO(PICOOD,HULSUB)
CALL MONOTE("TOTAL COMMAND NOT YET IMPLEMENTED")
KOND=" ":
RETURN
END
SUBROUTINE PICT03 & TOTAL TABULATIONS

(E H SCHLOSSER)

INCLUDE KOMKGT.LIST
EXTERNAL PIC000.NULSUB
CALL TRACE

CALL MVIAT0(PIC000,NULSUB)
CALL MNOTE("TOTAL COMMAND NOT YET IMPLEMENTED")
RETURN
END
SUBROUTINE PICTOS $ TOTAL TABULATIONS

(E. H. SCHLOSSER)

INCLUDE KOMIGT.LIST
EXTERNAL PICCONS.HULSUB
CALL TRACE

CALL NVTATO(PICCONS.HULSUB)
CALL MNOTE("TOTAL COMMAND NOT YET IMPLEMENTED")
RETURN
END
SUBROUTINE PICXGT  & INITIALIZATION ROUTINE FOR PICTAB

**HISTORY**

E H SCHLOSSER  LEC  11/21/75  ORIGINAL CODE
E H SCHLOSSER  LEC  01/23/79  ALLOW DEFAULT COMMANDS FROM MACDAM

**METHOD**

INITIALIZE PROGRAM. OPEN FILES. IDENTIFY SCENE. QUEUE DEFAULT COMMANDS.

**MACRO-DEPENDENT CODE**

UNIVAC EXEC-8 PROGRAM FILE NAMING CONVENTIONS.

**EXTERNAL REFERENCES**

- NVIA0: NAME NEXT 'VIA' & 'TO' SUBROUTINES
- PSTART: PROGRAM START INITIALIZATION
- OPEN3: OPEN FILE 3 (INPUT MSS OR RBV DATA)
- LDREON: LOAD EXACT OR NOMINAL REGISTRATION PARAMETERS
- IDLU3: IDENTIFY FILE 3 HARDWARE
- IDERTS: IDENTIFY MSS OR RBV DATA
- SYSADD: ADD DISK SYMBOLIC FILE OR ELT TO SYSIN RUNSTREAM
- MDFATL: SUBMIT FATAL DIAGNOSTIC MESSAGE

**EXCEPTIONS**

1. MISSING DEFAULT COMMANDS GENERATE A FATAL DIAGNOSTIC.

**GLOBAL DECLARATIONS**

INCLUDE KOMXGT.LIST  & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS

**LOCAL DECLARATIONS**

INTEGER LOCFIL  & LOCATION WITHIN DISK SYMBOLIC FILE ( IF > 0 )

L-155
C PROCEDURE
C -----------
C IDENTIFY PROGRAM
CALL PSTART( 'DAM PICTAB(0009)' )

ON RETURN, CALL PIC000 TO GET DEFAULT/USER COMMANDS
CALL NVIATO( PIC000.NULSUB)

OPEN FILES & LOAD REGISTRATION PARAMETERS
CALL OPEN3
CALL LOREGN
IF(MDATAC.NE.0) GO TO 300 8 DATA/CHECKOUT MODE

IDENTIFY ERTS SCENE
WRITE(6,225) 8 SKIP LINE
CALL IOLU3( 6)
CALL IDERTS( 6)

QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE
300 CALL SYSADD(LOCFIL. 'MACDAM','DEF-PICTAB'. '')
     IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL. 'DAM','DEF-PICTAB'. ' ')
     IF(LOCFIL.LE.0) CALL NDFATL( 'NO DEFAULT COMMANDS'

RETURN TO NEXT STATEMENT IN CALLING ROUTINE
RETURN
END
SUBROUTINE IOCPIC: IDENTIFY CURRENT COMMAND SPECS FOR PICTAB

1 UNIT: OUTPUT UNIT

HISTORY

E H SCHLOSSER  LEC 12/09/75  ALGORITHM CODING
J C CRISP  LEC 10/10/79  REVISE TO USE CHAR BUFFER Routines

METHOD

CHECK UNIT FOR 6 OR LEGAL ALTERNATE PRINT FILE. ENCODE FILE NAME FROM UNIT NUMBER. IDENTIFY SHARPENING SPECS, IF PRESENT, BUILD CHARACTER BUFFER, AND OUTPUT. IDENTIFY TRANSFORMATION SPECS, IF PRESENT, BUILD CHARACTER BUFFER, AND OUTPUT. BUILD CHARACTER BUFFER CONTAINING CHANNEL TYPE AND NUMBER, RADIANCE LIMIT VALUES AND SYMBOLS, SPACING, AND ORIGIN. OUTPUT BUFFER.

MACHINE-DEPENDENT CODE

UTILIZES UNIVAC EXEC B ER PRINTS AND ER PRNTAS

EXTERNAL REFERENCES

CINIT  # INITIALIZE CHARACTER BUFFER
C0CST  # CHARACTER BUFFER FOR CHARACTER STRING
C041N  # CHARACTER BUFFER FOR INTEGER
C041L  # CHARACTER BUFFER FOR REAL
CST4IN  # CHARACTER STRING FOR INTEGER
ERPRINT  # PRINT IMAGE ON TTY OR LINE PRINTER
ERPTA  # WRITE IMAGE TO ALTERNATE PRINT FILE
INTEGER LENCST  # LENGTH CHARACTER STRING
DOUBLE PRECISION C054IN  # VARIABLE LENGTH CHAR STRING FOR INTEGER

EXCEPTIONS

1. OUTPUT WILL BE TRUNCATED TO FIT THE LENGTH OF THE OUTPUT BUFFER.

2. IF UNIT IS NOT 6 OR A LEGAL ALTERNATE PRINT FILE (10*HALTM-1), THEN A FATAL ERROR IS GENERATED.

GLOBAL DECLARATIONS

-------------------

L-157
LOCAL DECLARATIONS

C

INTEGER NAMPIL, NRWCHN, NLCHAN, NPCHAN, NLC, MAXIS

CALL TRACE

PROCEDURE

Illuminate file name and character buffer

CALL CSTWIN(NAMFIL(1), (2)).(UNIT, 1)
CALL CBINIT (ICBUFI)

Check for valid output unit/print file

IF (UNIT.EQ.0. OR.
   UNIT.EQ.10.AND.UNIT.LE.(10+MATHM-1)) GO TO 200
   CALL MDFAIL (CBS4IN(UNIT, 1)). ' ' IS BAD UNIT IN IDCPIC')
   GO TO 900

Check for valid output unit/print file

200 CALL CB4CST (ICBUFI).

DO 250 NRWCHN=1,NRCH
   IF (.IRSF12(NRWCHN,1).EQ.0.AND.
      IRSF12(NRWCHN,2).EQ.0) GO TO 250
         NO COEFFS FOR CHAN
   IF (LENCST(ICBUFI, 2).EQ.0) CALL CB4CST (ICBUFI).
      "...
   CALL CB4CST (ICBUFI, 'SHA')
   CALL CB4IN (ICBUFI, NRWCHN, 2)
   CALL CB4CST (ICBUFI, 'SAM')
   CALL CB4RL (ICBUFI).
      FLOAT(IRSF12(NRWCHN, 2))2.***-12.4
   CALL CB4CST (ICBUFI, '.'
   CALL CB4RL (ICBUFI).
      FLOAT(IRSF12(NRWCHN, 2))2.***-12.4
290 CONTINUE
 CALL CB4CST (ICBUFI, '0')
 IF (LENCST(ICBUFI,3).LE.0) GO TO 290 & BLANK BUFFER
 IF (UNIT.EQ.6) CALL ERPRNT (1,2,ICBUFI)
 IF (UNIT.NE.6) CALL ERPRTA (NAMFIL,1,2,ICBUFI)

C

C OUTPUT TRANSFORMATION SPECS
C ---SPECS FOR LINEAR TRANSFORMATIONS
C
290 IF (IRTYP.EQ.'RAW') GO TO 400 & NO TRANSFORMATIONS
 DO 320 NLCHAN=1,2
 CALL CBINIT (ICBUFI)
 CALL CBW4ST (ICBUFI, '(LIN,')
 CALL CBW4IN (ICBUFI, NLCHAN,1)
 CALL CBW4CST (ICBUFI, 'HE',)
 DO 300 NRNCHAN=1,NRCHA
 CALL CB4RL (ICBUFI, RTLWGT(NRNCHAN,NLCHAN,1,1,4)
 CALL CB4CST (ICBUFI, '')
 CONTINUE
 CALL CB4CST (ICBUFI, 'GAIN,')
 CALL CB4RL (ICBUFI, RTLGCN(NLCHAN,1,3)
 CALL CB4CST (ICBUFI, 'BIAS,')
 CALL CB4RL (ICBUFI,
 FLOAT(LRT812(NLCHAN))42.00-12.1,31
 CALL CB4CST (ICBUFI, '1')
 IF (UNIT.EQ.6) CALL ERPRNT (1,2,ICBUFI)
 IF (UNIT.NE.6) CALL ERPRTA (NAMFIL,1,2,ICBUFI)
320 CONTINUE

C

C ---SPECS FOR POLAR TRANSFORMATIONS
C
 IF (IRTYP.NE.'POL') GO TO 400 & NO POLAR TRANSFORMATION
 CALL CBINIT (ICBUFI)
 CALL CBW4CST (ICBUFI, '1')
 DO 350 NPCHAN=1,2
 IF (NPCHAN.EQ.2) CALL CB4CST (ICBUFI, '...')
 CALL CB4CST (ICBUFI, 'POL,')
 CALL CBW4IN (ICBUFI, NPCHAN,1)
 CALL CBW4CST (ICBUFI, 'GAIN,')
 CALL CBW4RL (ICBUFI,
 FLOAT(NRT812(NPCHAN))*2.**-12.1,3)
 CALL CBW4CST (ICBUFI, 'BIAS,')
 CALL CBW4RL (ICBUFI,
 FLOAT(NRT824(NPCHAN))*2.**-24.1,3)
 CONTINUE
 CALL CB4CST (ICBUFI, '1')
 IF (UNIT.EQ.6) CALL ERPRNT (1,2,ICBUFI)
 IF (UNIT.NE.6) CALL ERPRTA (NAMFIL,1,2,ICBUFI)

C

C IDENTIFY CHANNEL TYPE AND NUMBER(S)
C
400 CALL CBINIT (ICBUFI)
 CALL CBW4CST (ICBUFI, '(CHAN,')
 CALL CBW4CST (ICBUFI, IRTYP,1,1,1,3)
DO 490 NLCH=1.NLIMCH
   CALL CB4CST (ICBUFI, 'IC
   CALL CB4IN (ICBUFI, LINCH(NLCH))
490 CONTINUE

C IDENTIFY RADIANCE LIMIT VALUES AND SYMBOLS FOR FIRST CHANNEL
C
   CALL CB4CST (ICBUFI, 'RAD.'1)
   CALL CB4IN (ICBUFI, LCVL0(1,1))
   CALL CB4CST (ICBUFI, ' ')
   CALL CB4CST (ICBUFI, KSYM(LCVLO(1,1),1,1))
   CALL CB4IN (ICBUFI, LCVM(1,1))
   CALL CB4CST (ICBUFI, ' ')
   CALL CB4CST (ICBUFI, KSYM(LCVMI(1,1),1,1))

C IDENTIFY RADIANCE LIMIT VALUES FOR ANY OTHER CHANNELS
C
   IF (NLIMCH.LT.2) GO TO 600
   DO 550 NLCH=2,NLIMCH
      CALL XRE077(2) A=2 *** DUMP REG X2 *** A=2
      CALL CB4CST (ICBUFI, ' ')
      CALL CB4IN (ICBUFI, LCVL0(NLCH,1))
      CALL XRE077(2) A=2 *** DUMP REG X2 *** A=2
      CALL CB4IN (ICBUFI, LCVM(NLCH,1))
   550 CONTINUE

C IDENTIFY SPACING
C
   600 CALL CB4CST (ICBUFI, 'SPA')
   DO 650 NAXIS=1,2
      CALL CB4CST (ICBUFI, ' ')
      SPA=FLOAT(MSA0W(NAXIS,WSPI100))/100.
      IF (SPA.EQ.AINT(SPA)) CALL CB4IN (ICBUFI, IFIX(SPA,1))
      IF (SPA.NE.AINT(SPA)) CALL CB4RL (ICBUFI, SPA,1,2)
   650 CONTINUE

C IDENTIFY ORIGIN
C
   CALL CB4CST (ICBUFI, 'ORIG.SCAN.')
   CALL CB4IN (ICBUFI, MSA0W(NLIN,WOR10))
   CALL CB4CST (ICBUFI, ' ')
   CALL CB4IN (ICBUFI, MSA0W(NSAM,WOR10))
   CALL CB4CST (ICBUFI, ' ')

L-160
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

C OUTPUT CHANNEL/RADIANCE/SPACING/ORIGIN INFO
C
IF (UNIT.EQ.6) CALL ERPRINT (1,22,ICBUF1)
IF (UNIT.NE.6) CALL ERPRTA (NAMFIL,1,22,ICBUF1)

C NORMAL RETURN
C
900 RETURN
END
SUBROUTINE OPRPIC  & OPEN ALTERNATE PRINT FILES FOR PICTAB

HISTORY

E H SCHLOSSER  LEC  06/28/79  ORIGINAL CODE
J C CRISP  LEC  11/19/79  PICDEF FOR KPAONI & MALTHI

METHOD

A maximum of MALTHI (from PICDEF) alternate print files are opened and initialized.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

OPENPR  OPEN ALTERNATE PRINT FILES

EXCEPTIONS

1. One print file is always opened. Unless in data/checkout mode.
2. If MALTH is greater than MALTHI (from PICDEF), it will be set to MALTHI.
3. If KPAON is greater than KPAOHI (from PICDEF), it will be set to KPAOHI.

GLOBAL DECLARATIONS

INCLUDE KOMXGT.LIST  # COMMON PROGRAM EXECUTION SWITCHES.COUNTERS
INCLUDE PICDEF.LIST  # PARAMETERS FOR PICTAB

LOCAL DECLARATIONS

NONE

PROCEDURE
C --------
C CALL TRACE
C
C NO MORE THAN MALTHI ALTERNATE PRINT FILES FOR PICTAB
C
MALTH=MAX0(MALTH,1)
MALTH=MIN0(MALTH,MALTHI)
C
C NO MORE THAN KPAGEI COLUMN.S PER PAGE FOR PICTAB
C
KPAGE=MAX0(KPAGE,1)
KPAGE=MIN0(KPAGE,KPAONI)
C
C OPEN FILE(S)
C
IF(MDATC.NE.0) GO TO 900  & DATA/CHECKOUT MODE
CALL OPENPR
C
C IDENTIFY ERTS SCENE
C
WRITE(10,225)  & SKIP LINE
CALL IOLU3(10)
CALL IDERTS(10)
C
C 900 RETURN
END
CONTROL

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROGRAM CONTROL
-------------

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

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

THIS PROGRAM IS LIMITED TO THAT PART OF THE WORLD COVERED BY THE
CLARKE 1966 SPHEROID (NORTH AMERICA).

UP TO 350 POINTS MAY BE USED. POINTS WITH USER-ASSIGNED NUMBERS
BETWEEN +1 AND +999 ARE CONTROL POINTS AND WILL BE USED IN THE
ADJUSTMENT. POINTS WITH NUMBERS BETWEEN -1 AND -999 ARE CHECK POINTS
AND WILL NOT BE USED IN THE ADJUSTMENT. RESIDUAL ERRORS
FOR CONTROL POINTS AND CHECK POINTS ARE COMPUTED SEPARATELY.
AT LEAST 5 CONTROL POINTS MUST BE USED. IF ONE NETWORK IS TO
BE USED FOR ALL 4 STRIPS OF A SCENE IT SHOULD CONTAIN AT LEAST 6
CONTROL POINTS. WITH AT LEAST 1 CONTROL POINT IN EACH OF THE 4
CCT STRIPS.

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

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

MACHINE-DEPENDENT CODE
-----------------------
NONE.

EXTERNAL REFERENCES
--------------------
NVIAITO & NAME 'VIA' 'TO' ROUTINES
VIATO & CALL 'VIA' 'TO' ROUTINES
VIA TO
EXTERNAL CONGOO. CONXOT

EXTERNAL CONGOO. CONXOT

EXCEPTIONS

1. IF THE SCENE, GEOMETRY, OR ATTITUDE COMMANDS ARE MISSING OR HAVE ANY INVALID SPECIFICATIONS, THEN CONTROL WILL REFUSE TO ADJUST THE NETWORK.

2. IF FEWER THAN 5 VALID POINTS ARE PROVIDED, THEN CONTROL WILL REFUSE TO ADJUST THE NETWORK.

3. IF THE RMS (ROOT MEAN SQUARE) ERROR OF THE ADJUSTED NETWORK IS GREATER THAN 400 METERS, THEN ON EXIT, CONTROL WILL GENERATE A "FATAL ERROR" AND ABORT.

4. IF CONXOT DOES NOT CALL NVIATO TO CHANGE THE 'VIA' AND/OR 'TO' ROUTINES, THEN CONTROL WILL CALL TO CONXOT IN AN ENDLESS LOOP!

GLOBAL DECLARATIONS

- INCLUDE KONXOT.LIST: COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
- INCLUDE KOMLOO.LIST: COMMON LOG FILE BUFFER, I/O PKT. POINTERS
- INCLUDE KOMLUJ.LIST: COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
- INCLUDE KOMNER.LIST: COMMON CRS SCENE PARAMETERS
- INCLUDE KOMFIT.LIST: COMMON ADJUSTMENT/REGISTRATION PARAMETERS
- INCLUDE KOMNET.LIST: COMMON CONTROL NETWORK POINT COORDINATES

PROCEDURE

- CALL NVIATO( CONGOO, CONXOT): A FIRST CALL IS VIA CONGOO TO CONXOT

CONTINUE

CALL VIATO

GO TO 100

END

STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE
PROGRAM CONTROL/VIRTUAL

HISTORY

E N SCHLOSSER LEC 09/02/74 ORIGINAL CODE
E N SCHLOSSER LEC 11/06/79 SNAP.FZINT: NO 'N' IN DEPEND!

METHOD

CONSTRUCT SNAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT BGT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE SNAP & BGT COMMANDS TO TEMPORARY FILE PO.
ADD TEMPORARY FILE BO. TO RUNSTREAM.

MACHINE-DEPENDENT CODE

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

EXTERNAL REFERENCES

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

EXCEPTIONS

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

GLOBAL DECLARATIONS

1. PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER 44 AS FOLLOWS:
   1 2 = REAL TIME
   1 3 = LOW EXEC
   1 4 = DEPEND
   1 5 = DEADLINE BATCH
   1 6 = BATCH

1. BGT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER AS IN
I M A S T E R D I T B O T T O N A T I O N.

LOC A L D E C L A R A T I O N S

-------------

A X S

$001. D-B A N K
SSH FORM 6.6.10

| LABEL | 100.1. "F" | 0.0.0.0 | LABEL: I WD. FORTRAN. FIELDATA
|-------|-----------|--------|-------------------------|
| N AMPS | $SSH 0.0.0.0 | DATA. 0 MSG. | FIELDATA
| NAMING | "$XQTS: MAP.FZM CAM.CONTROL-MAP.CONTROL | "$XQTS":
| ADDO S | $SSH 0.0.0.0 | ADD | CAM: SYS-HAPOPT | "$XQTS":
| $XQTS | $SSH 0.0.0.0 | $XQTS: | CAM: | "$XQTS":
| SFCSD | $00.0.0.0 | END-OF-FILE | STOP WORD | 0.
| PP | FORM 12.6.10 | "BASE.T 20.
| CSFA00 | "ADD 20.
| SAVRE0 | RES 1 | "20": MS 33. LASSDF. "0" 0.

PROCEDURE

-------------

$001. L-BANK

| CONTROL | LA | A0. | " | A0 = " | " | |
|---------|---|-----|--------|----------|-----------|
| TNCU | A4.9 | SKIP NEXT INST IF A4<9 4 (NOT DEMAND)
| SA, SP | A0.$NAPING+2 | DEMAND: BLANK OUT N OPTION
| LA | A0. (CSFAO0) | ADDRESS OF $AS0 IMAGE
| ER | CSF8 | DO IT
| SA | A0. SAVRE0 | STORE 8
| PNP | PRINT | IF P F S. I. SAVRE0; PRINT $AS0 STATUS

GETOPT | LOAD OPT LTRS INTO A2.A3.A4

PUTOPT | D5 | A2.XQTING+2 | STORE OPTION LETTERS INTO $XQTS IMAGE (3 WORDS -- MAX 18 OPT LETTERS)
| SA | A4.XQTING+4

WRITE | LA | A0. (IOPK1) | ADDRESS OF I/O PACKET
| ER | 10N8 | WRITE SFDF IMAGES TO 20.

ADD | LA | A0. (CSFADD) | ADDRESS OF $ADD IMAGE
| ER | CSF8 | DO IT
| ER | EXITS

END | CONTROL

L-169
CONTROL-OVERLAY STRUCTURE

HISTORY

E N SCHLOSSER  LEC  03/19/75  ORIGINAL CODE
E N SCHLOSSER  LEC  07/14/78  CHANGE OVERLAYS TO REDUCE THRASHING
E N 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.

SEQ S-MAIN
IN DAM.CONTROL/  MAIN PROGRAM
IN DAM.NVIAO    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.COM000   CALL USER-SPECIFIED PHASE 0 ROUTINE
IN DAM.NTABSDAM DAM UNIT & TABLE GOES IN SAME SEG W/ FORTRAN I/O

PHASE 0.1.2.9 COMMANDS (FORTRAN I/O ALLOWED)

SEQ S-XQTEXI*
IN DAM.COMXQT   CONTROL INITIALIZATION ROUTINE
IN DAM.COMEXI   CONTROL TERMINATION ROUTINE

SEQ S-KMD---S-XQTEXI*
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.KMDECO   GET/CHECK GEOMETRY
IN DAM.KMDF    BEGIN IF...FI BLOCK
IN DAM.KMDNEW   PRINT NEWS
IN DAM.KMDEX   CONDITIONALLY PERFORM NEXT COMMAND
IN DAM.KMDOFF   TURN OFF MODE SWITCHES
IN DAM.KMDOM    TURN ON MODE SWITCHES
IN DAM.KMDPEE   PEEK AT LABELLED COMMON
IN DAM.KMP01    GET CONTROL/CHECK/QUERY POINT
IN DAM.KMP0K    POKE AT LABELLED COMMON
IN DAM.KMDSC    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.KMDOXX   MACRO COMMANDS
IN DAM.KMDOZON  GET/CHECK UTM PROJECTION ZONE
IN DAM.KMDOAD   DYNAMIC ADD

L-189
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

SEQ S-CONADJ*.S-XQTEXI
IN OAH.CONADJ . ADJUST NETWORK

SEQ S-CONDIA*.S-XQTEXI
IN OAH.CONDIA . DIAGRAM NETWORK/ERRORS
SUBROUTINE CON000  CALL PHASE 0 SUBROUTINES FOR CONTROL
1 NAMSUB  NAME OF SUBROUTINE TO CALL (OR NULSUB)

C HISTORY
C-------
C
E H SCHLOSSER LEC  07/07/75 ORIGINAL CODE
E H SCHLOSSER LEC  01/24/79 MACRO COMMANDS & TIME COMMAND
J C CRISP LEC  12/15/79 PEEK/POKE/GEOMETRY/CENTER/SIZE/IF/IF

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

C MACHINE-DEPENDENT CODE
------------------------
NONE.

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

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

C GLOBAL DECLARATIONS
---------------------
INCLUDE NULCST.LIST  DEFINE NULL CHARACTER STRING

C LOCAL DECLARATIONS
------------------
INTEGER KMD  FIRST 3 CHARS OF USER COMMAND (BLANK AFTER DCE)
MAIN PROGRAMS/Routines

INTEGER LSSTAT 8 READS STATUS ('EOF' MEANS END-OF-FILE)
INTEGER KASE 8 MODIFIED 1-C-E OF FIRST CHAR OF COMMAND

* *
* *
* *
*

C

C PROCEDURE
C------------
C
C CALL PREVIOUSLY NAMED SUBROUTINE
C
C CALL TRACE
C CALL NULSUB 8 CALL TO NULSUB DOES NOTHING
C
C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C
KOMO=' NUL' 8 IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
CALL READ5(LSSTAT,NULCST) 8 FILL UNIT 5 BUFFER. NO CUE MESSAGE
IF(LSSTAT.NE.0) KOMO='EXI'
IF(KOMO.NE.'EXI') CALL GET5AL(KOMO,13), NULCST) 8 GET 3 ALPHA CHARS
C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C
KASE=ICE(KOMO)-ICE('A')+1 8 A TO Z = 1 TO 26
C
C CASE STATEMENT ON MODIFIED ICE OF COMMAND'S FIRST CHARACTER
C
1 IF(KASE.LT.1.OR.(KASE.GT.26)) KASE=27 8 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) 6 KASE
C
C DETERMINE COMMAND. PERFORM COMMAND. CHANGE KOMO TO BLANK
C
401 CONTINUE 8*** A
IF(KOMO.EQ.'ADJ') CALL CONAdj(KOMO) 8 ADJUST
IF(KOMO.EQ.'ATT') CALL KMOATT(KOMO) 8 ATTITUDE
GO TO 800
C
402 CONTINUE 8*** B
GO TO 800
C
403 CONTINUE 8*** C
IF(KOMO.EQ.'CEN') CALL KMOcen(KOMO) 8 CENTER
IF(KOMO.EQ.'CLE') CALL KMOCLE(KOMO) 8 CLEAR
GO TO 800
C
404 CONTINUE 8*** D
IF(KOMO.EQ.'DIA') CALL CONDIA(KOMO) 8 DIAGRAM
GO TO 800
C

L-173
405 CONTINUE 4**** E
   IF(KOMD.EQ. 'EXI') CALL CONEXI(KOMD)   8 EXIT
   IF(KOMD.EQ. 'EXP') CALL KMODEXP(KOMD)  8 EXPLAIN
   GO TO 800
C  
406 CONTINUE 4**** F
   IF(KOMD.EQ. 'FI') CALL KMODFI(KOMD)  8 FI (ENDIF)
   GO TO 800
C  
407 CONTINUE 4**** G
   IF(KOMD.EQ. 'GEO') CALL KMODGEO(KOMD)  8 GEOMETRY
   GO TO 800
C  
408 CONTINUE 4**** H
   GO TO 800
C  
409 CONTINUE 4**** I
   IF(KOMD.EQ. 'IF') CALL KMODIF(KOMD)  8 IF
   GO TO 800
C  
410 CONTINUE 4**** J
411 CONTINUE 4**** K
412 CONTINUE 4**** L
413 CONTINUE 4**** M
   GO TO 800
C  
414 CONTINUE 4**** N
   IF(KOMD.EQ. 'NEW') CALL KMODNEW(KOMD)  8 NEWS
   IF(KOMD.EQ. 'NEXT') CALL KMODNEXT(KOMD)  8 NEXT
   GO TO 800
C  
415 CONTINUE 4**** O
   IF(KOMD.EQ. 'OFF') CALL KMODOFF(KOMD)  8 OFF
   IF(KOMD.EQ. 'ON') CALL KMODON(KOMD)  8 ON
   GO TO 800
C  
416 CONTINUE 4**** P
   IF(KOMD.EQ. 'PEE') CALL KMODPEE(KOMD)  8 PEEK
   IF(KOMD.EQ. 'POI') CALL KMODPOI(KOMD)  8 POINT
   IF(KOMD.EQ. 'PO') CALL KMODPO(KOMD)  8 POKE
   IF(KOMD.EQ. 'POK') CALL KMODPOK(KOMD)  8 POKE
   GO TO 800
C  
417 CONTINUE 4**** Q
418 CONTINUE 4**** R
   GO TO 800
C  
419 CONTINUE 4**** S
   IF(KOMD.EQ. 'SCE') CALL KMODSCE(KOMD)  8 SCENE
   IF(KOMD.EQ. 'SIZ') CALL KMODSIZ(KOMD)  8 SIZE
   GO TO 800
C  
420 CONTINUE 4**** T
   IF(KOMD.EQ. 'TIM') CALL KMODTIM(KOMD)  8 TIME
   GO TO 800

L-174
481 CONTINUE 8**** U
482 CONTINUE 8**** V
483 CONTINUE 8**** W
484 CONTINUE 8**** X
485 CONTINUE 8**** Y
  GO TO 800
C
486 CONTINUE 8**** Z
   IF(KOMD.EQ.'ZOM') CALL KMDZOM(KOMD)   ZONE
   GO TO 800
C
487 CONTINUE 8**** NOT ALPHABETIC
   IF(KOMD.EQ.'SAD') CALL KMDSAD(KOMD)   SADD
   IF(KOMD.NE.'') CALL KMDPOI(KOMD)   POINT
C
C IF COMMAND WAS NOT FOUND, TRY MACRO-COMMAND
C
800 IF(KOMD.NE.'' ) KOMD='CON--' 3ST 3 CHAR OF PROG NAME PLUS --'
   IF(KOMD.NE.'' ) CALL KMDXXX(KOMD)   MACRO COMMAND HANDLER
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
   IF(KOMD.NE.'' ) CALL WARNSI('INVALID COMMAND --')
C
C RETURN TO MAIN FOR CALL VIA/IO NAMED SUBROUTINE IN ANY OVERLAY
C
RETURN
END
SUBROUTINE CONADJ: ADJUST CONTROL NETWORK
U KOND: 0 1: FIRST 3 CHAR OF COMMAND 0: SPACES

METHOD

ESTIMATE SCENE TRANSVERSE MERCATOR (STM) CENTRAL MERIDIAN AND SATELLITE ALTITUDE.
TRANSFORM NON-LINEAR, SKEW SCANNER COORDINATES OF CONTROL POINTS TO LINEAR, SKEW CORRECTED COORDINATES. TRANSFORM EARTH COORDINATES OF CONTROL POINTS TO SCENE TRANSVERSE MERCATOR (STM) COORDINATES.
PERFORM INITIAL LEAST SQUARES ADJUSTMENT BETWEEN CONTROL POINT COORDINATES IN THESE TWO COORDINATE SYSTEMS.
REFINE SCENE TRANSVERSE MERCATOR CENTRAL MERIDIAN AND SATELLITE ALTITUDE.
PERFORM FINAL LEAST SQUARES ADJUSTMENT.

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

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GCONST: GET GEOMETRIC CONSTANTS FOR REGISTRATION
MDHARN: PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
MDUNIT: PRINT PAGE HEADING
O4V: UTM (OR STM) COORDINATES FOR GEOGRAPHIC COORDINATES
Q4V: GEOGRAPHIC COORDINATES FOR UTM (OR STM) COORDINATES
DLSTSQ: DOUBLE PRECISION LEAST SQUARES ADJUSTMENT
REVERT: COMPUTE COEFFICIENTS OF INVERSE TRANSFORMATION
STREOB: STORE REGISTRATION PARAMETERS ON UNIT 0
DBL4CS: DOUBLE PRECISION CBS4CS & VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH

CONADJ
001

L-176
1. Invalid values for any of the following generate 'Warning' diagnostics:
   - Scene number
   - Samples per scene
   - Pitch
   - Roll
   - Number of control points
   - Geometry

2. An extra specification generates a 'Warning' diagnostic.

Global declarations

Include

- COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
- COMMON ENTS SCENE PARAMETERS
- COMMON ADJUSTMENT/REGISTRATION PARAMETERS
- COMMON CONTROL NETWORK COORDINATE TABLE
- DEFINE STRUCTURE OF WINDOW PACKETS
- DEFINE COORDINATE TRANSFORMATION FUNCTIONS
- DEFINE NULL CHARACTER STRING

Local declarations

Integer NDSave  # Temporary save for contents of ND Totl on entry

Call Trace

Check for valid scene and geometry

NDSave = ND Totl
   IF((NRTS(11).LT.11).OR.
     (NRTS(11).GT.5)) CALL MDWARNI('SCENE NOT DEFINED')
   IF((NERGO. NE. 'ERT').AND.
     (NERGO. NE. 'HOM').AND.
     (NERGO. NE. 'LCC').AND.
     (NERGO. NE. 'PS').AND.
     (NERGO. NE. 'SOM').AND.
     (NERGO. NE. 'UTH')) NERGO = 'BAD'
   IF((NERGO. EQ. 'BAD'). AND.
     (NERSAM. LT. 3300)) NERGO = 'ERT'  # Temp patch for old b1p format scenes
   IF((NERGO. EQ. 'BAD')) CALL MDWARNI('GEOMETRY NOT DEFINED')

Check 'ERT' attitude and size

IF(NERGO. NE. 'ERT') GO TO 130
IF((ABS(PITDEG).LT.9.1).OR.(ABS(ROLDEG).LT.9.1)) CALL HWARN1
  = 'ATTITUDE NOT SPECIFIED WITH GEOMETRY'
IF((NERSAM.LT.100).OR.(NERSAM.GT.10000)) CALL HWARN1
  = 'SIZE (SAMPLES) NOT DEFINED'
go to 180

C CHECK NON-"CRT" ATTITUDE, SIZE, AND CENTER
C 190 IF((ABS(PITDEG).LT.9.1).AND.(ABS(ROLDEG).LT.9.1)) go to 190
     PITDEG=0.
     ROLDEG=0.
  160 IF((ABS(PITDEG)+ABS(ROLDEG)).NE.0.) CALL HWARN1
    = 'NON-ZERO ATTITUDE WITH', 'CBS4COSNIPEREG.1111.(311). GEOMETRY'
IF((NERSAM.LT.100).OR.(NERSAM.GT.10000)) CALL HWARN1
    = 'SIZE (SAMPLES) NOT DEFINED'
IF((CTRLIN.LT.90.1).OR.(CTRLIN.GT.9000.1)) CALL HWARN1
    = 'CENTER (LINE) NOT DEFINED'
IF((CTRSAM.LT.90.1).OR.(CTRSAM.GT.9000.1)) CALL HWARN1
    = 'CENTER (SAMPLE) NOT DEFINED'

C CHECK NUMBER OF CONTROL POINTS
C 190 IF(INCTLPT.LT.91) CALL HWARN1; 'NOT ENOUGH CONTROL POINTS'
    IF((NOSAVE.NE.NOTOL).AND.
    & (MBATCH.NE.0)) CALL HDFA1; 'INCOMPLETE NETWORK'

C DETERMINE WHICH POINTS TO PRINT RESIDUAL ERRORS FOR
Ccerypoint = 'ALL'
CALL GETSKM(NPNT,3, NULCN)
IF(NPNT.EQ.'CON') GO TO 310
     & JUST CONTROL POINTS
IF(NPNT.EQ.'CHE') GO TO 320
     & JUST CHECK POINTS
IF(NPNT.EQ.'ALL') GO TO 330
     & ALL POINTS
IF(NPNT.EQ.'NON') GO TO 340
     & NONE
CALL WARN1; 'BAD ADJUST SPECIFICATION - -'
go to 900

310 NPPO VS=1
     NPRNE=0
IF(INCTLPT.GT.0) GO TO 400
     CALL HWARN1; 'NO CONTROL POINTS'
go to 900

320 NPPO VS=0
     NPRNE=1
IF(INCTLPT.GT.0) GO TO 400
     CALL HWARN1; 'NO CHECK POINTS'
go to 900

330 NPPO VS=0
     NPRNE=0
     GO TO 400

340 NPPO VS=0
     NPRNE=0

L-170
C DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C
C INITIALIZE
C
400 IF (INDSAVE .NE. NOTOTO) GO TO 900
NWDOW=ABS(NWDOW)
NERSAT(1)=
NERSAT(1)=
ALTV=0.
ALTSAM=0.
CALL OCONST
ROLRAD=ROLCEO*0.017932921
NWDOW=ABS(NWDOW)
CALL HDUNIT( 0.8)
WRITE(6,415) NERTS
415 FORMAT(' SCENE ',11,J4,':',15)

C
C ESTIMATE CONTROL NETWORK CENTROID, COVERAGE, AND STM CENTRAL MERIDIAN
C
CTOLIN=0.
CTOSAM=0.
CTOLAT=0.
CTOLON=0.
DO 610 K=1,NETHI
IF (NETPT(K).LE.0) GO TO 610
CTOLIN=CTOLIN+ADJNET(NLIN,K)
CTOSAM=CTOSAM+ADJNET(NSAM,K)
CTOLAT=CTOLAT+ADJNET(NLAT,K)
CTOLON=CTOLON+ADJNET(NLON,K)
610 CONTINUE
CTOLIN=CTOLIN/FLOAT(NCTLP)
CTOSAM=CTOSAM/FLOAT(NCTLP)
CTOLAT=CTOLAT/FLOAT(NCTLP)
CTOLON=CTOLON/FLOAT(NCTLP)
STMHO=CTOLON
CALL PCTCOV(PCTCTRL)

C
C SAVE NOMINAL ALTITUDE
C
SANTK=M(NER,ALTSM/(TAN(ROLRAD+SCNTHW)
& -TAN(ROLRAD+SCNTHW))

C
C ESTIMATE CORRECTED LINE AND SAMPLE NUMBERS AND STM COORDINATES
C
CALL TAPCOR

C
C ESTIMATE FORWARD STM COEFFICIENTS
C
CALL DLSTSQ(NETHI.NETPT,
8 CORNET(NLIN,11,CORNET(NSAM,11).
8 STMNET(NGO,11),STMNET(NGO,11),CORSTM)
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C COMPUTE FINAL ALTITUDE AND PROJECTION CENTRAL MERIDIAN
C
IF(NEQEO.EQ.'ERT') CALL SATALT(ALTTHN.SAMKH)
CALL O4ICTRLAT.CTRLON.CTRILN.CTRSAM)
STMCHD=AIN(0.5+100.*CTRLON1/100).
IF(NEQEO.EQ.'UTH') STMCHD=(IFIX(CTRLON1)+6)/6+3 8 CLOSEST UTH C.M.

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

C COMPUTE FINAL FORWARD & INVERSE STM COEFFICIENTS
C
CALL DLTSQ(NETHI,NETPT,0.TBNET(KLIN,11),TBNET(KSAM,11),
& STMNET(WNO,11),STMNET(HEA,11),CORSTM)
CALL REVERT(CORSTM,STMCOR)

C COMPUTE STM/GEODETIC COORDINATES OF SCENE CENTER AND NADIR
C
CTRN=STMCH4(CTRLIN.CORS4A(CTRLIN.CTRSAM1))
CTRE=STMCH4(CTRLIN.CORS4A(CTRLIN.CTRSAM1))
CALL O4I(CTRLAT.CTRLON.CTRE.CTRN.STMCHD)
DIRLAT=0. 8 FUTURE COMPUTATION
DIRLON=0. 8 FUTURE COMPUTATION

C COMPUTE STM/GEODETIC COORDINATES OF CONTROL NETWORK CENTROID
C
CTON=STMCH4(CTOLIN.CORS4A(CTOLIN.CTOSAM1))
CTDE=STMCH4(CTOLIN.CORS4A(CTOLIN.CTOSAM1))
CALL O4I(CTDLAT.CTOLON.CTDE.CTON.STMCHD)

C COMPUTE DIFFERENCE BETWEEN SCENE CENTER & CONTROL NETWORK CENTROID
C
OCLIN=CTRLIN-CTOLIN
OCSAM=CTRSAM-CTOSAM
OCLAT=CTRLAT-CTOLAT
OCLON=CTRLON-CTOLON
CKM=.001*SQRT((CTRE-CTDE)**2+(CTRN-CTON)**2)

C COMPUTE ANGLE OF ROTATION FROM STM TO SRM
C (EASTING AXIS OF SRM IS PARALLEL TO MSS SCAN LINES)
C
ROTAD=ATAN2(CORSTM(2)/CORSTM(3))

C COMPUTE 1:1 MAPPING COEFFICIENTS IN METERS
C
SRLIN=CORSTM(1)+CORLIN(CR(2)*CORSAM+CRI(3))
C

DOAII PACKAGE APPENDIX L
MAIN PROGRAMS/ROUNIES

C SRSAM=CORSAM(C1+COS(ROTAD))
C CORSAM(1)=CORSAM(C1)*SIN(ROTAD)-CORSAM(1)*COS(ROTAD)
C CORSAM(2)=0.
C CORSAM(3)=0.
C CORSAM(4)=CORSAM(C1)*COS(ROTAD)+CORSAM(1)*SIN(ROTAD)
C CORSAM(5)=CORSAM(C1)*COS(ROTAD)+CORSAM(2)*SIN(ROTAD)
C CORSAM(6)=0.

C PRINT ADJUSTMENT SUMMARY
C
WRITE(6,619)
1 CTRLIN.CTRLSAM.CTRLLAT.CTRLON.
2 CTOLIN.CTOSAM.CTOLAT.CTOLON.
3 DCLIN.DOCSAM.DCLAT.DCOLON.DCKN
619 FORMAT('0.20X,'LINE SAMPLE LATITUDE LONGITUDE'/
1 'SCENE CENTER '2(F9.2,1X),2(F10.5,1X)/
2 'CONTROL CENTROID '2(F9.2,1X),2(F10.5,1X)/
3 'CENTER-CENTROID '2(F9.2,1X),2(F10.5,1X)/
4 '('F5.1,'KHM')'
WRITE(6,629)STCMD
629 FORMAT('PROJECTION CENTRAL MERIDIAN '10X,F10.5)
WRITE(6,635)PCTCTL
635 FORMAT('CONTROL COVERAGE 'F6.1,'PERCENT')
WRITE(6,649)ALTNOM
649 FORMAT('NONOMAL ALTITUDE 'F6.1,'KM')
WRITE(6,655)ALTMN
655 FORMAT('COMPUTED ALTITUDE 'F6.1,'KM')

C
C Compute and Print Residual Errors
C CALL ERROR(FRMSMET)
C
C Store Registration Parameters
C CALL STORE
MNNON=MNNDI=1

C Restore Nominal Altitude
C ALTOM=ALTNI IM

C Normal Return
C 900 KMD= 1
RETURN
MAIN PROGRAMS/ROUTINES

INTERNAL SUBROUTINE TAPCOR

CALL TRACE
ALTERM=ALTERM+SAMIKM
DO 180 K=1,NETH
CONET(MLN.K)=CONET(MLN.K)+ADJMET(MLN.K).
CONET(HSAM.K)=CONET(HSAM.K)+ADJMET(HSAM.K)
180 CONTINUE

INTERNAL SUBROUTINE SATALT(ALTkm,SAMIKM)

CALL TRACE
BEGIN=STNMT(N*MERLIN.I.]
BEGIN=STNMT(N*MERLIN.I.]
END=STNMT(N*MERLIN.I.]
END=STNMT(N*MERLIN.NERSAM)
SAMIKM=BASEK'M(.O01*SNCE(BEGIN-END)*2 + (BEGIN-END)*2)
SAMIKM=BASEK'M/(TANR0LRAD*SNCTH2-TANR0LRAD*SNCTH2)
RETURN

INTERNAL SUBROUTINE ERRORS(RMSNET)
REAL RLCOR(R),RSCOR(R),RLAT(2),RLON(2),RUTH(2)
CALL TRACE

INITIALIZE
DO 180 K=1,2
RLCOR(K)=0.
RSCOR(K)=0.
RLAT(K)=0.
RLON(K)=0.
RUTH(K)=0.
180 CONTINUE
WRITE(6,215)
215 FORMAT1
1 'RESIDUAL ERRORS IN CONTROL POINT ADJUSTMENT':
2 'GPOINT LINE SAMPLE LATITUDE LongITUDE',
3 'METERS POINT')

C COMPUTE RESIDUAL ERRORS

DO 806 K=1,NETHI
   NPRINT=NPRPOS
   NODTYP=''
   IF(NETPT(K).EQ.0) GO TO 802
   KRL=0
   NPRINT=NPRNEO
   NODTYP='CHK'
802 ELCOR=CORNET(WLIN.K)-CORNET(WLIN.K)+STMNET(WEA.K).STMNET(WNO.K)
   ESCOR=CORNET(WSAM.K)-CORNET(WSAM.K)+STMNET(WEA.K).STMNET(WNO.K)
   RLCOR(KRL+1)=RLCOR(KRL+1)+ELCOR*2
   RSCOR(KRL+1)=RSCOR(KRL+1)+ESCOR*2
   CMPE=STMNET(WLIN.K).CORNET(WSAM.K)
   CMPN=STMNET(WSAM.K).CORNET(WLIN.K)
   CALL GNUICMPLAT.CMPLON.CMPE.CMPN.STMCMOI
   ELAT=STMNET(WLAT.K)-CMPLAT
   ELON=STMNET(WLON.K)-CMPLON
   EUTM=(STMNET(WEA.K)-CMPE)*0.2+(STMNET(WNO.K)-CMPN)*0.2
   RLAT(KRL+1)=RLAT(KRL+1)+ELAT*2
   RLON(KRL+1)=RLON(KRL+1)+ELON*2
   RUTM(KRL+1)=RUTM(KRL+1)+EUTM
   EUTM=SORT(EUTM)
   IF(NPRINT.NE.0) WRITE(6,804)
      NETPT(K).ELCOR.ESC0R.ELAT.ELON.
   804 FORMAT(2X,14.2X.2(F7.2,1X).2(F10.5,1X).2X.F7.5,1X.A3.14)
   806 CONTINUE

C COMPUTE ROOT MEAN SQUARE ERRORS

NODTYP='CTL'
   N=NCTLPT
   KRL=1
807 RLCOR(KRL+1)=SORT(1RLCOR(KRL+1)/N)
   RSCOR(KRL+1)=SORT(RSCOR(KRL+1)/N)
   RLAT(KRL+1)=SORT(RLAT(KRL+1)/N)
   RLON(KRL+1)=SORT(RLON(KRL+1)/N)
   RUTM(KRL+1)=SORT(RUTM(KRL+1)/N)
   WRITE(6,809) NODTYP,RLCOR(KRL+1),RSCOR(KRL+1),
      RLAT(KRL+1),RLON(KRL+1),RUTM(KRL+1)
   809 FORMAT(1XM='A3.2(F7.2,1X),2(F10.5,1X),F8.1,1X,13.1X,13.1X.A3.14.2X,'1PTS'))
   IF(KRL.EQ.0) GO TO 810
   IF(NCTLPT.EQ.NETHI) GO TO 810 \ & NO CHECK POINTS
   NODTYP='CHK'
   N=NETHI-NCTLPT \ & NUMBER OF CHECK POINTS
   KRL=0
   GO TO 807

L-183
INTERNAL SUBROUTINE PCTCOV(PCTCTL)
C THIS INTERNAL SUBROUTINE ESTIMATES THE PERCENT OF THE SCENE COVERED BY THE CONTROL NETWORK. THE AREA OF THE NETWORK IS APPROXIMATED BY THE AREA OF AN ELLIPSE WITH RADII EQUAL TO GEOMETRIC MEAN RADII OF THE TWO SMALLEST ROTATED ENVELOPES.
REAL RLS(4), OLSMIN(2), OLSMAX(2)
REAL PI/3.141592651
CALL TRACE
C
C INITIALIZE
DO 140 N=1,31.2
RLS(N,1)=999999.  A INITIAL MINIM.
140 RLS(N+1,1)=-999999. A INITIAL MAXIHA
OLSMIN(1)=999999.
OLSMIN(2)=999999.
OLSMAX(1)=999999.
OLSMAX(2)=999999.
C
C FIND ROTATED ENVELOPES
DO 240 K=1,NETHI
IF(NETPT(K).LT.0) GO TO 280
ROTLIN=ADJNET(MLIN.K)
ROTSAM=ADJNET(WSAM.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=R0TLINEOS(P1/16.1)-ROTSAM*SIN(P1/16.1)
ROTLIN=TEMP
240 CONTINUE
280 CONTINUE
C
C FIND MINIMUM/MAXIMUM ENVELOPE DIMENSIONS
DO 320 NR=2,32.2
OLSMIN(1)=AMINI(OLSMIN(1),(RLS(NR,1)-RLS(NR-1,1)))
OLSMAX(1)=AMAXI(OLSMAX(1),(RLS(NR,1)-RLS(NR-1,1)))
320 CONTINUE
DO 340 NR=2,32,2
IF((RLS(NR,1)-RLS(NR-1,1)).EQ.OLSHMIN(1)) GO TO 330
OLSHMIN(2)=AMIN1(OLSHMIN(1), (RLS(NR,1)-RLS(NR-1,1)))
330 IF((RLS(NR,1)-RLS(NR-1,1)).EQ.OLSHMAX(1)) GO TO 340
OLSHMAX(2)=AMAX1(OLSHMAX(1), (RLS(NR,1)-RLS(NR-1,1)))
340 CONTINUE

C ESTIMATE RADII/AREA/PCT
C
PIXNW=PI
& *SORT(OLSHMIN(1)+OLSHMIN(2)/4)
& *SORT(OLSHAX(1)+OLSHAX(2)/4)
PIXTOT=FLOAT(NERLIN)+FLOAT( NERSAM)
PCTCTL=100.*PIXNW/PIXTOT
RETURN
END
SUBROUTINE CONOIA 8 DIAGRAM CONTROL NETWORK
U KOND) & 1: FIRST 3 CHAR OF COMMAND O: SPACES

C HISTORY

C

E H SCHLOSSER LEC 08/03/73 ORIGINAL CODE
E H SCHLOSSER LEC 12/02/75 ALPHANUMERIC COMMAND
E H SCHLOSSER LEC 07/19/76 DELETE RETURN K
E H SCHLOSSER LEMSCO 05/27/80 CHECK SIZE, CENTER

C METHOD

PRINT POINT NUMBERS (LESS SIGN AND HUNDREDS DIGIT) AT APPROXIMATE
LOCATIONS WITHIN SCENE.

C MACHINE-DEPENDENT CODE

DIMENSION AND FORMAT SPECIFICATIONS ASSUME 6 CHARACTERS PER WORD.

C EXTERNAL REFERENCES

C

MDWARN 2 PRINT/LOG/COUNT 'WARNING' DIAGNOSTIC MESSAGE
MDUNIT 2 PRINT PAGE HEADING
DIAERR 2 DIAGRAM ERRORS IN CONTROL NETWORK

C EXCEPTIONS

1. INVALID VALUES FOR ANY OF THE FOLLOWING GENERATE 'WARNING' DIAGNOSTICS:
   SCENE SIZE (LINES)
   SCENE SIZE (SAMPLES)
   SCENE CENTER (LINE)
   SCENE CENTER (SAMPLE)
   TYPE OF POINTS TO DIAGRAM
   NUMBER OF POINTS

2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

C GLOBAL DECLARATIONS

C

INCLUDE KOMXQ LIST 2 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMN R LIST 2 COMMON ERTS SCENE PARAMETERS
INCLUDE KOMFIT LIST 2 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMNET LIST 2 COMMON CONTROL NETWORK COORDINATE TABLE

L-106
C

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

INCLUDE WINGEF.LIST  # DEFINE STRUCTURE OF WINDOW PACKETS (6 DEFINES)
INCLUDE NULCST.LIST  # DEFINE NULL CHARACTER STRING

C LOCAL DECLARATIONS

C

INTEGER NOSAVE  # 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/
   11111222222333334444
   1 'CONTROL POINTS ONLY
   2 'CHECK POINTS ONLY
   3 'ALL POINTS

C PROCEDURE

C CALL TRACE

C

CHECK IF SCENE SIZE AND CENTER ARE DEFINED

NOSAVE=NOTOTL
IF((NERLIN.LT.100).OR.(NERLIN.GT.1000)) CALL MDWARN;
   'SIZE (LINES) NOT DEFINED'
IF((NERSAM.LT.100).OR.(NERSAM.GT.1000)) 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(NOSAVE.NE.NOTOTL) GO TO 900

C DETERMINE WHICH POINTS TO DIAGRAM

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 WARN('BAD DIAGRAM SPECIFICATION ---')
   GO TO 900

210 ASSIGN 130 TO 1POS
   ASSIGN 130 TO 1NEG
   NODT='1'
   IF(NCTLPT.GT.0) GO TO 300
   CALL MDWARN('NO CONTROL POINTS')
   GO TO 900

220 ASSIGN 130 TO 1NEG
   ASSIGN 370 TO 1POS

L-187
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

NODT=8
IF(METHI-NCTLPT).GT.0) GO TO 300
   CALL MDWARN ("NO CHECK POINTS")
   GO TO 900
230 ASSN 330 TO IPOS
ASSN 330 TO INC0
NODT=3
IF(METHI.GT.0) GO TO 300
   CALL MDWARN ("NO CHECK POINTS")
   GO TO 900
240 CALL DIAERR
   GO TO 900

C
C INITIALIZE

300 NWNDW=1ABS(NWNDW)
IF(MDAC.NE.0) GO TO 900 8 DATA/CHECKOUT MODE
   CALL HUNIT(0.6)
   WRITE(6,305) NERTS, (NODTYP(N),NODT), N=1,4)
305 FORMAT (SCENE 'II.J4.,-‘J5. ' -- '4AB)
   DO 320 I=1,884
   320 MIXL(1,1)=0

C
C LOAD DIAGRAM ARRAY WITH POINT NUMBERS

DO 370 N=1,NETHI
   IF(INETPT(N)) INEO,370,1POS
330 INTEMP=ABS(NETPT(N)) 8 DON'T PRINT SIGN
   INTEMP=MOD(INTEMP,100) 8 STRIP OFF HUNDREDS DIGIT
   ILMAP=N-30:*ADJNET(ILIN,N)=(CTRLIN+NERLIN)/2.
   ILMAP=MAX(0,ILMAP,2)
   ILMAP=MIND(ILMAP,35)
   ISMAP=1+24.*ADJNET(ISAM,N)=(CTRLAM+NERSAM)/2.
   ISMAP=MID(ISMAP,24)
   IF(MIXL(ISMAP,ILMAP).EQ.0) GO TO 360
   IF(MIXL,0T.181) GO TO 360
   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).EQ.0) GO TO 370
360 MIXL(ISMAP,ILMAP)=INTEMP
370 CONTINUE

C
C

L-189
CONDIA

MAIN PROGRAMS/Routines

C DIAGRAM POINTS ABOVE SCENE
C
  MATT(01)='(2H .'
  MATT(06)='1X .'
  MATT(15)='1X .'
  MATT(22)='1X .'
  MATT(29)='1X .'
  ILMAPN=1
  MLAX=3
  CALL TROLIN

C DIAGRAM POINTS WITHIN SCENE
C
  WRITE(6,505)
  505 FORMAT(' ')
  CALL TROLIN
  WRITE(6,505)

C DIAGRAM POINTS BELOW SCENE
C
  MATT(01)='(2H .'
  MATT(06)='1X .'
  MATT(15)='1X .'
  MATT(22)='1X .'
  MATT(29)='1X .'
  ILMAPN=34
  MLAX=36
  CALL TROLIN
  WRITE(6,505)
  505 FORMAT(' ')
  CALL TROLIN
  WRITE(6,505)

900 KOMO=0
  RETURN

SUBROUTINE TROLIN
  DO 800 ILMAP=ILMAPN,MLAX
  GO TO 700
  DO 700 ISMAP=1,2
  MFI=ISMAP+1+(ISMAP-1)/6
  IF(MFXL(ISMAP,ILMAP).EQ.0) GO TO 800
  MATT(MFI)='1Z .'
  GO TO 800
  700 CONTINUE
  GO TO 800
  800 CONTINUE
GO TO 700
600 MATT(NF1)='AR,' MIXL(ISMAP,ILMAP)= '
700 CONTINUE
WRITE(6,MATT) (MIXL(ISMAP,ILMAP),ISMAP=1,24)
800 CONTINUE
RETURN
END
SUBROUTINE CONEXI & TERMINATION ROUTINE FOR CONTROL

-E H SCHLOSSER

INCLUDE KOMXQT.LIST  & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST  & DEFINE NULL CHARACTER STRING

CALL TRACE

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

PSTOP

TERMINATE PROGRAM

IF(INDFATL.NE.0) CALL PABORT(NULCST)
CALL PSTOP(NULCST)

PSTOP DOES NOT RETURN

END
SUBROUTINE CONXOT & INITIALIZATION ROUTINE FOR CONTROL

HISTORY

E N SCHLOSSER LEC 11/22/79 ORIGINAL CODE
E N SCHLOSSER LEC 01/27/79 ALLOW DEFAULT COMMANDS FROM MACDAM

METHOD

INITIALIZE PROGRAM, INITIALIZE PITCH & ROLL, QUEUE DEFAULT COMMANDS.

MACHINE-DEPENDENT CODE

UNIVAC EXEC-B PROGRAM FILE NAMING CONVENTIONS.

EXTERNAL REFERENCES

NAME NEXT 'VIA' & 'TO' SUBROUTINES
PROGRAM START INITIALIZATION
ADD DISK SYMBOLIC FILE OR ELT TO SYSIN RUNSTREAM
SUBMIT FATAL DIAGNOSTIC MESSAGE

EXCEPTIONS

MISSING DEFAULT COMMANDS GENERATE A FATAL DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE COMMONER.LIST COMMON ERTS SCENE PARAMETERS
PITCH IN DEGREES
ROLL IN DEGREES
EXTERNAL COMMONER NULSUB

LOCAL DECLARATIONS

LOCATION WITHIN DISK SYMBOLIC FILE ( IF > 0 )

PROCEDURE
IDENTIFY PROGRAM

CALL PSTARTI 'DAM CONTROL(8009)'

FLAG PITCH & ROLL AS UNKNOWN

PITDEC=-999999.
ROLDEC=-999999.

QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE

300 CALL SYSADD(LOCFIL, 'MACDAM', 'DEF-CONTROL', '')
   IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL, 'DAM', 'DEF-CONTROL', '')
   IF(LOCFIL.LE.0) CALL MOPATL('NO DEFAULT COMMANDS')

ON RETURN, CALL CON000 TO GET DEFAULT/USER COMMANDS

CALL NVIATOI CON000.NULSUBI

RETURN TO NEXT STATEMENT IN CALLING ROUTINE

RETURN

END
Subroutine DIAERR - Diagram Control/Check Point Errors

This subroutine diagrams errors in both control and check points.

Determine if mode is batch or demand

100 IF(BATCH.EQ.0) IMDE=35
    IF(IMDE.EQ.102) GO TO 202

Set plot constants for demand mode

300 BETA=18.0
    ICTR=18
    NCOL=33
    GO TO 204

Set plot constants for batch mode

202 BETA=48.
    ICTR=48
    NCOL=98
    204 ITEMP='JUNK'

Determine if line or sample error to be plotted

CALL GETSKM(ITEMP,3,0)
    IF(ITEMP.EQ.'LIN') GO TO 210
    IF(ITEMP.EQ.'SAM') GO TO 220
    CALL WARN('BAD ERROR AXIS ---')
    GO TO 900

210 LSPRT=1
    JERRAX='LINE'
    GO TO 300
**MAIN PROGRAMS/ROUTINES**

220 L$PRT=2
   JERRAX='SAMPLE'

SET UP SORT

230 CALL GETS$N(I$EMP,2,-0)
   IF(I$EMP.EQ.'S0R') GO TO 230
   CALL WARM$(SORT NOT SPECIFIED --')
   GO TO 290

290 CALL GETS$N(I$EMP,3,-0)
   IF(I$EMP.EQ.'LIN') GO TO 410
   IF(I$EMP.EQ.'SAM') GO TO 420
   CALL WARM$(BAD SORT AXIS --')
   GO TO 290

410 L$SRRT=1
   J$RTAX='LINE'
   GO TO 450

420 L$SRRT=2
   J$RTAX='SAMPLE'

CHECK IF NETWORK HAS BEEN ADJUSTED

450 IF(RMS$ET.NE.0) GO TO 390
   CALL MDWAR$NETWORK NOT ADJUSTED')
   GO TO 390

590 DO 600 K=1,N$ETHI
   ITAG(K)=FIX(AD$JMET(L$SRRT.K)*2.820)
   ASM$ET(I$TAG(K))=K
   CONTINUE

BEGIN SORT

   CALL ISRTN$ITAG(N$ETHI))
   IF(IN$DE.EQ.35) GO TO 610
   GO TO 64

610 WRITE(6,505) JERRAX,JS$RTAX

505 FORMAT(1'POINT',3X,'ERROR',3X, 'SORT',3X,'+'2.X,4X.AB,2X.AB)
   GO TO 617

614 WRITE(6,515) JERRAX,JS$RTAX

515 FORMAT(1'POINT',3X,'ERROR',3X,'SORT',3X, '2.X,4X.AB,2X.AB)
   GO TO 617

617 IN$=2.0

CALCULATE AND PLOT THE ERRORS

700 DO 790 K=1,N$ETHI
   KK=ASM$ET(I$TAG(K))
   NOD$YP='*'
   IF(NETPT(KK).GT.0) GO TO 702

L-199
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

END

ORIGINAL PAGE IS OF POOR QUALITY
SUBROUTINE OLSTSQ(NODES, IPT, X, Y, P, Q, A)
C (W O EPPLEER / E M 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 OOJR TO SOLVE THE SIMULTANEOUS EQUATIONS REPRESENTED BY
C MATRIX H (OOJR 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 IPT(1)=IPT(1)
CALL TRACE
C
NTOT=NODES+2
NCTLPT=0
X=0.
Y=0.
P=0.
Q=0.
DO 1 K=1, NTOT, 2
IF(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(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(X, P, IPT, NTOT)
H(2, 4)=DAIP(Y, P, IPT, NTOT)
H(3, 4)=DAIP(Z, P, IPT, NTOT)
H(1, 5)=DAIP(X, Q, IPT, NTOT)
H(2, 5)=DAIP(Y, Q, IPT, NTOT)
H(3, 5)=DAIP(Z, Q, IPT, NTOT)
V(1)=4
CALL DQRH(1,1,3.3.3S7.JC.V)
A(1)=M(1,4)
A(2)=M(2,4)
A(3)=M(3,4)-A(1)*X8-A(2)*Y8+PB
A(4)=M(1,5)
A(5)=M(2,5)
A(6)=M(3,5)-A(4)*X8-A(5)*Y8+QB
DO 3 K=1,NTOT.2
IF(IPT(K/2+1).LT.0) GO TO 3
X(K)=X(K)+X8
Y(K)=Y(K)+Y8
P(K)=P(K)+PB
Q(K)=Q(K)+QB
3 CONTINUE
GO TO 9
7 WRITE(6,8) JC(1)
8 FORMAT(1,OVERFLOW AFTER ROW*12)
9 RETURN

FUNCTION OAIP(R,S,IP,NTOT)
DIMENSION R(II,S(II),IP(I)
DOUBLE PRECISION SUM
SUM=0.
DO I=1,NTOT.2
IF(IPT(K/2+1).LT.0) GO TO 1
SUM=SUM+DBLE(R(K))*DBLE(S(K))
1 CONTINUE
OAIP=SUM
RETURN
END
PROGRAM CLASSIFY

HISTORY

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

METHOD

This program classifies data from a Landsat 'X', 'AM', or 'PM' MSS tape assigned to unit 3 and stores the classified but unresampled data in a random access detection file for use in the same or subsequent runs by other programs in the DAM package.

The user specifies the name and spectral limits for one material, a window defining the area to be classified, and the type of detection file to generate.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

NVIATO & NAME 'VIA' 'TO' ROUTINES
VIATO & CALL 'VIA' 'TO' ROUTINES
VIA TO EXTERNAL CLAXOT, CLAXOT

EXCEPTIONS

1. If control has not been executed in the current run prior to CLASSIFY and satisfactorily adjusted a control network for the scene to be processed by CLASSIFY, then CLASSIFY will refuse to process the scene and will generate a 'FATAL ERROR.'

2. If CLASSIFY encounters any fatal errors (such as fatal tape parity/positioning errors) it will error terminate and flag the output detection file as defective. The run will not be aborted, but subsequent executions of PRDENS/PRCLASS/PLTCLASS will be alerted to this fatal error.

3. If CLAXOT does not call NVIATO to change the 'VIA' and/or 'TO' routines, then CLASSIFY will call to CLAXOT in an endless loop!
GLOBAL DECLARATIONS

INCLUDE KOMXQ7.LIST  a COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMLOG.LIST  a COMMON LOG FILE BUFFER, I/O PKT. POINTERS
INCLUDE KOMLU3.LIST  a COMMON POINTERS/FLAGS FOR UNIT 3
INCLUDE KOM LuS.LIST  a COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
INCLUDE KOMLEN.LIST  a COMMON I/O PKTS FOR DETECTION FILES (UNITS 2N
INCLUDE KOMIMW.LIST  a COMMON INPUT WINDOW PACKETS
INCLUDE KOMOWW.LIST  a COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMNER.LIST  a COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLs.LIST  a COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST  a COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMIRT.LIST  a COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
INCLUDE KOMDET.LIST  a COMMON DETECTION FILE WINDOW PACKETS & DATES
INCLUDE KOMSLM.LIST  a COMMON SPECTRAL LIMITS
INCLUDE KOMALT.LIST  a COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS

PROCEDURE

CALL NVIATO(  CLA000,CLAXQT)  a FIRST CALL IS VIA CLA000 TO CLAXQT
   100 CONTINUE
   CALL VIATO
   CALL NVIATO(  CLA000,CLAXQT)  a FIRST CALL IS VIA CLA000 TO CLAXQT
   GO TO 100
   END  a (STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE)
PROGRAM CLASSIFY/VIRTUAL

HISTORY

E N SCHLOSSER LEC 08/02/74 ORIGIHAL CODE
E N SCHLOSSER LEC 11/06/79 $MAP.FIN: 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 8 FUNCTION TO SUBMIT EXEC-8 CONTROL STATEMENT
ER IOUS 8 INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS 8 TERMINATE PROGRAM EXECUTION
DAM.CLASSIFY-MAP & SYMBIQL MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
DAM.SYS-MAPOPT 8 STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS $ASG-O & $PREP-O.

GLOBAL DECLARATIONS

(PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER A# AS FOLLOWS:
1 2 = REAL TIME
1 3 = LOW EXEC
1 4 = DEMAND
1 5 = DEADLINE BATCH
1 6 = BATCH

&XQT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER AS IN

L-202
MASTER BIT NOTATION.

LOCAL DECLARATIONS

AXRS

$100 D-BANK

SSSH FORM 6.6.6.18

LABSDF SSSH 050.1.4".0 LABEL I WD. FORTRAN FIELDATA

LABING "SOPPFM"

MAPSDF SSSH 000.9.0.0 DATA. 9 WDS. FIELDATA

MAPING ":XQTS: MAP.F2N DAM.CASSIFY-MAP.CASSIFY :XQTS"

ADDOSDF SSSH 000.9.0.0

ADDING ":XQTS: ADD DAM.SYS-HAPOPT :XQTS"

XQTSDF SSSH 000.9.0.0

XQTING ":XQTS: XQT.1 CASSIFY :XQTS"

EOFSDF - 0 END-OF-FILE STOP WORD

PF FORM 12.6.18

CSFAST '

CSFADD 

SAVREO RES 1

IOPK T1500 '20'. WS 33.LABSDF.'0' 0

PROCEDURE

-----------

$101 I-BANK

CLASSIFY L.A.U A0 .

TNE.U A4..4 SKIP NEXT INST IF A4<4 (NOT DEMAND)

SA.SS A0.MAPlN+2 DEMAND BLANK OUT N OPTION

LA A0.(CSFAST) ADDRESS OF BASO IMAGE

ER CSF$ DO IT

SA A0.SAVREO STORE &

PRINT (PF 2.1.SAVREO) PRINT BASO STATUS

GETOPT ' LOAD OPT LTRS INTO A2.A3.A4

PUTOPT OS A2.XQTING+2 STORE OPTION LETTERS INTO XQT IMAGE

SA A4.XQTING+4 (3 WORDS -- MAX 18 OPT LETTERS)

WRITE LA A0.(IOPK T) ADDRESS OF I/O PACKET

FR 10WS WRITE SDF IMAGES TO 20.

ADD LA A0.(CSFADD) ADDRESS OF SADD IMAGE

ER CSFS DO IT

ER EXITS

END CLASSIFY
CLASSIFY OVERLAY STRUCTURE

HISTORY

- E N SCHLOSSER LEC 08/23/74 ORIGINAL CODE
- E N SCHLOSSER LEC 07/14/78 CHANGE OVERLAYS TO REDUCE THRASHING
- E N SCHLOSSER LEC 10/30/78 TOLERANCE COMMAND
- E N SCHLOSSER LEC 01/30/79 MACRO COMMANDS & TIME COMMAND
- E N SCHLOSSER LEC 12/19/79 PEEK, POKE, IF, FI
- MARY TOMPKINS LEMSCO 01/11/80 CHANGE OVERLAYS

LIB DAM.

SEQ S-MAIN
- IN DAM.CLS:OVERLAYS MAIN PROGRAM
- IN DAM.NVIAO NAME/CALL 'VIA' AND 'TO' SUBROUTINES
- IN DAM.NLCSUB DO NOTHING
- IN DAM.NS-BLOCK . BLOCK DATA SUBROUTINE

UTILITIES FOR MAIN

SEQ S-FLINFO* . (S-MAIN)
- IN DAM.FRINFO . GET FILE DESCRIPTIVE INFORMATION

SEQ S-R3TASC* . (S-MAIN)
- IN DAM.CSTVARS CHARACTER STRING FOR ASCII
- IN DAM.R3TREC READ ONE RECORD FROM TAPE (UNIT 3)

MONITOR FOR PHASE 0.1.2.9 COMMANDS

SEQ S-CLA0129* . (S-FLINFO.S-R3T.SC)
- 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

SEQ S-READS* . (S-CLA0129)
- IN DAM.READS 'READ' INTO UNIT 5 BUFFER
- IN DAM.GETS 'GET' FREE-FORMAT FIELD FROM UNIT 5 BUFFER
- IN DAM.WRNO PROCESS WARNING DIAGNOSTIC FOR UNIT 5 FIELD
- IN DAM.SPANS ENABLE/DISABLE SPANNING FOR UNIT 5

SEQ S-OPNCLPR* . (S-CLA0129)
- IN DAM.OPRCLA . OPEN ALT PRRT FILE
- IN DAM.CLOSPP . CLOSE ALT PRRT FILE

SEQ 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-OPENCLPR.S-CALCROP)
IN DAM.CLAXQT    . CLASSIFY INITIALIZATION ROUTINE
IN DAM.CLAXEI    . CLASSIFY 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 PXBDEF 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-03TDRMDR*. (S-OP3MDP)
IN DAM.03TDR    . 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-LORE08*. (S-XQTEX1)
IN DAM.LDRE08    . LOAD LOCATION PARAMETERS FROM UNIT 8

SEQ S-CLOSTOP*. (S-XQTEX1)
IN DAM.CLOSE3    . CLOSE INPUT SCAN DATA FILE (UNIT 3)
IN DAM.CL020N    . CLOSE OUTPUT DETECTION FILE (UNIT 2N)
IN DAM.PSTOP     . GENERAL TERMINATION ROUTINE

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

L-205
PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CLASSIFY-MAP

IN DAN.KNOEXP . EXPLAIN PROGRAM/COMMAND
IN DAN.KNDF1 . END IF...FI BLOCK
IN DAN.KNDIF . BEGIN IF...FI BLOCK
IN DAN.KNDFROM . PRINT NEWS
IN DAN.KNDFROMX . CONDITIONALLY PERFORM NEXT COMMAND
IN DAN.KNDOFF . TURN OFF MODE SWITCH(ES)
IN DAN.KNDOFF1 . TURN ON MODE SWITCH(ES)
IN DAN.KNDFEE . PEEK
IN DAN.KNDFPO . POKE
IN DAN.KNDFRE . RENUMBER (GET/CHECK NEW WINDOW SEQUENCE NUMBER)
IN DAN.KNDFTIM . PRINT CLOCK TIME & CHARGE TIME

SEO S-GEOMETRY*. (S-READS, S-OPNCLPR, S-CALCROP)
IN DAN.KNWIN . GET/CHECK WINDOW ENVELOPE/VERTICES
IN DAN.KDOFIN . GET/CHECK UTH PROJECTION ZONE

SEO S-SPECS-9*. (S-READS, S-OPNCLPR, S-CALCROP)
IN DAN.KDCHMA . GET/CHECK RAW/TRANSFORMED SCANNER CHANNEL(S)
IN DAN.KENDAE . GET/CHECK PAGE HEADING(S)
IN DAN.KENDNAM . GET/CHECK TRANSFORMATION/MATERIAL NAME
IN DAN.KENDORI . GET/CHECK WINDOW ORIGIN
IN DAN.KENDRAD . GET/CHECK RADIANCE LIMITS
IN DAN.KENDTOL . GET/CHECK TOLERANCE

SEO S-MISC*. (S-READS, S-OPNCLPR, S-CALCROP)
IN DAN.KENOP . GET/CHECK NUMBER OF OUTPUT COPIES
IN DAN.KENDLIN . GET/CHECK LINEAR TRANSFORMATION WEIGHTS/GAIN/BIAS
IN DAN.KENDPAO . SKIP TO TOP OF NEXT PAGE
IN DAN.KENDPRI . GET/CHECK PRINTER SPECIFICATIONS
IN DAN.KENDPOL . GET/CHECK POLAR TRANSFORMATION GAIN/BIAS
IN DAN.KENDSHA . GET/CHECK SHARPENING FILTER COEFFICIENTS

SEO S-EXEC*. (S-READS, S-OPNCLPR, S-CALCROP)
IN DAN.KNDXXX . MACRO COMMANDS
IN DAN.KNDOAD . DYNAMIC SADD
IN DAN.KNDOAS . DYNAMIC SBSO
IN DAN.KNDOBR . DYNAMIC SBRKPT
IN DAN.KNDOFR . DYNAMIC SFREE
IN DAN.KNDOLO . DYNAMIC SLOG

SEO S-CLADET*. (S-READS, S-OPNCLPR, S-CALCROP)
IN DAN.CLADET . DETECT RADIANCE/DENSITY/CLASS (PHASE 0)

MONITOR FOR PHASE 3.4.5 COMMANDS ----------------------------------------

SEO S-CLA345*. S-CLA0129
IN DAN.CLA34S

UTILITIES FOR PHASE 3.4.5 COMMANDS ---------------------------------------

SEO S-RO330P*. (S-CLA34S)
IN DAN.RO33IL . READ MSS DATA IN BIL FORMAT (UNIT 3)
IN DAN.RO33BSQ . READ MSS DATA IN BSO FORMAT (UNIT 3)

SEO S-RO331P*. (S-CLA34S)

L-206
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM.RO3BIP . BBS DATA IN BIP FORMAT (UNIT 3)

SEQ S-RO3DSKNUl*. (S-CLA345)
IN DAM.RO3DSK . DATA ON DISK IN PX80DEF FORMAT
IN DAM.RO3NUL . SYNTHETIC DATA -- NO UNIT 3

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

SEQ S-CLADE3*. (S-RO3MOP.S-RO3BIP.S-RO3DSKNUl) IN DAM.CLADE3 . DETECT RADIANCE IN SELECTED CHANNEL

SEQ S-CLADE4*. (S-RO3MOP.S-RO3BIP.S-RO3DSKNUl) IN DAM.CLADE4 . DETECT BINARY CLASSIFICATION DENSITY

SEQ S-CLADES*. (S-RO3MOP.S-RO3BIP.S-RO3DSKNUl) IN DAM.CLADES . DETECT CLASS USING PARTITION TABLE
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM, CLASSIFY/VIRTUAL

CLASSIFY-HAP/VIRTUAL 001

L-208
SUBROUTINE CLA006: CALL USER-SPECIFIED PHASE 0 SUBROUTINES FOR CLASSIFY

HISTORY

- C E N SCHLOSSER LEC 07/1/73 ORIGINAL CODE
- C E N SCHLOSSER LEC 08/1/70 DELETE RETN K'S & NUMERIC OPTIONS
- C E N SCHLOSSER LEC 10/30/79 TOLERANCE COMMAND
- C E N SCHLOSSER LEC 01/30/79 MACRO Commands & TIME COMMAND
- C E N SCHLOSSER LEC 11/30/79 PEEK, POKE, IF, FI

METHOD

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

MACHINE-DEPENDENT CODE

- NONE.

EXTERNAL REFERENCES

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

EXCEPTIONS

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

GLOBAL DECLARATIONS

- INCLUDE NULCS111ST.LIST: DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

L-209
INTEGER KOMO  \ A FIRST 3 CHAR OF USER COMMAND (BLANK AFTER DONE
INTEGER LSTAT  \ A READS STATUS ("COF" MEANS END-OF-FILE)
INTEGER KASE  \ A MODIFIED 1-C-E OF FIRST CHAR OF COMMAND

PROCEDURE
----------

CALL PREVIOUSLY NAMED SUBROUTINE

CALL TRACE
CALL NAMSUB \ A CALL TO NULSUB DOES NOTHING

READ COMMAND FROM UNIT 9 (CARD READER OR TERMINAL)

KOMO=NUL' \ A IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
CALL READS(LSTAT,  \ A FILL BUFFER, BLANK CUE MESSAGE
IF(LSTAT.NE.' ') KOMO= 'EXI'
IF(KOMO.NE.'EXI') CALL GETSAL(KOMO,13), NULST) \ A GET 3 ALPHA CHARS

CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT

KASE=ICF(KOMO)-ICE('A')+1 \ A TO Z = 1 TO 26

CASE STATEMENT ON MODIFIED ICE OF COMMAND'S FIRST CHARACTER

IF(KASE.LT.11.OR.(KASE.GT.26)) KASE=27 \ A NOT ALPHA
00 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
8 .kase

DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOMO TO BLANK

401 CONTINUE 4*** A
402 CONTINUE 4*** B
00 TO 800

403 CONTINUE 4*** C
IF(KOMO.EQ.'CHA') CALL KNOCH'A(KOMO) \ A CHANNEL
IF(KOMO.EQ.'CLE') CALL KHOCE'I(KOMO) \ A CLEAR
IF(KOMO.EQ.'COPIES') CALL KNOEPE(KOMO) \ A COPIES
00 TO 800

404 CONTINUE 4*** D
IF(KOMO.EQ.'DET') CALL CLADE'I(KOMO) \ A DETECT
00 TO 803

405 CONTINUE 4*** E
GAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IF(KOMD.EQ."EXI") CALL CLAEXI(KOMD) 8 EXIT
IF(KOMD.EQ."EXP") CALL KNDEXP(KOMD) 8 EXPLAIN
GO TO 800
C
406 CONTINUE 8*** F
IF(KOMD.EQ."FI") CALL KMDFI(KOMD) 8 FI (ENDIF)
GO TO 800
C
407 CONTINUE 8*** G
GO TO 800
C
408 CONTINUE 8*** H
IF(KOMD.EQ."HEA") CALL KMDHEA(KOMD) 8 HEADING
GO TO 800
C
409 CONTINUE 8*** I
IF(KOMD.EQ."IF") CALL KMDIF(KOMD) 8 IF
GO TO 800
C
410 CONTINUE 8*** J
411 CONTINUE 8*** K
GO TO 800
C
412 CONTINUE 8*** L
IF(KOMD.EQ."LIN") CALL KMDLIN(KOMD) 8 LINEAR
GO TO 800
C
413 CONTINUE 8*** M
GO TO 800
C
414 CONTINUE 8*** N
IF(KOMD.EQ."NA") CALL KMDNA(KOMD) 8 NAME
IF(KOMD.EQ."NE") CALL KMDNE(KOMD) 8 NEWS
IF(KOMD.EQ."NX") CALL KMDNX(KOMD) 8 NEXT
GO TO 800
C
415 CONTINUE 8*** O
IF(KOMD.EQ."OFF") CALL KMDOFF(KOMD) 8 OFF
IF(KOMD.EQ."ON") CALL KMDON(KOMD) 8 ON
IF(KOMD.EQ."OR") CALL KMDOR(KOMD) 8 ORIGIN
GO TO 800
C
416 CONTINUE 8*** P
IF(KOMD.EQ."PA") CALL KMDPA(KOMD) 8 PAGE
IF(KOMD.EQ."PE") CALL KMDPE(KOMD) 8 PEEK
IF(KOMD.EQ."PO") CALL KMDPO(KOMD) 8 POKE
IF(KOMD.EQ."POL") CALL KMDPOL(KOMD) 8 POLAR
IF(KOMD.EQ."PRI") CALL KMDPRI(KOMD) 8 PRINTER
GO TO 800
C
417 CONTINUE 8*** Q
GO TO 800
C
418 CONTINUE 8*** R
IF(KOMD.EQ."RAD") CALL CLARAD(KOMD) 8 RADIANCE
GO TO 800
C
C-2

L-211
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C 419 CONTINUE 8*** S
  IF(KOMD.EQ.'SHA') CALL KMDSHA(KOMD) 8 SHARPENING
  GO TO 800

C 420 CONTINUE 8*** T
  IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD) 8 TIME
  IF(KOMD.EQ.'TOL') CALL KMDTOL(KOMD) 8 TOLERANCE
  GO TO 800

C 421 CONTINUE 8*** U

C 422 CONTINUE 8*** V
  GO TO 800

C 423 CONTINUE 8*** W
  IF(KOMD.EQ.'WIN') CALL KMDWIN(KOMD) 8 WINDOW
  GO TO 800

C 424 CONTINUE 8*** X

C 425 CONTINUE 8*** Y
  GO TO 800

C 426 CONTINUE 8*** Z
  IF(KOMD.EQ.'ZON') CALL KMDZON(KOMD) 8 ZONE
  GO TO 800

C 427 CONTINUE 8*** NOT ALPHABETIC
  IF(KOMD.EQ.'BAD') CALL KMDBAD(KOMD) 8 BADD
  IF(KOMD.EQ.'BAD') CALL KMDBAD(KOMD) 8 BADD
  IF(KOMD.EQ.'SAS') CALL KMDSAS(KOMD) 8 SASG
  IF(KOMD.EQ.'SBR') CALL KMDSBR(KOMD) 8 SBRKPT
  IF(KOMD.EQ.'SFR') CALL KMDSFR(KOMD) 8 SFREE
  IF(KOMD.EQ.'SLO') CALL KMDSLO(KOMD) 8 SLOG

C IF COMMAND WAS NOT FOUND, TRY MACRO-COMMAND
C 800 IF(KOMD.NE. '1') KOMD='CLA- ' 8 1ST 3 CHAR OF PROG NAME PLUS ' -'
  IF(KOMD.NE. '1') CALL KMXX(KOMD) 8 MACRO COMMAND HANDLER

C COMMAND IS INVALID IF STILL NOT FOUND
C 800 IF(KOMD.NE. '1') CALL WARN('INVALID COMMAND - -')

C FORCE ALL FORTRAN I/O ROUTINES INTO SEQ WITH CLA000 (NEVER PERFORMED)
C IF(KOMD.EQ.'JUNK') READ(895,895) KMD
  895 FORMAT(1X)

C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C RETURN
END
SUBROUTINE CLAIM 1 CALL PREVIOUSLY NAMED SUBROUTINE FOR CLASSIFY
1 NAME OF SUBROUTINE TO CALL

C CALL PREVIOUSLY NAMED SUBROUTINE
C CALL TRACE
C CALL NAMSUB

C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C RETURN
C END
SUBROUTINE CLA351
& CALL PHASE 3/4/5 SUBROUTINES FOR CLASSIFYING
& NAME OF SUBROUTINE TO CALL

C (E N SCHLOSSER)
C CALL PREVIOUSLY NAMED SUBROUTINE
C CALL TRACE
C CALL NAMSUB
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C RETURN
C END
SUBROUTINE CLADET: 0 INITIATE DETECTION & GENERATION OF DETECTION FILE

U KOND1 0 I: FIRST 3 CHAR OF COMMAND O: SPACES

HISTORY

E H SCHLOSSER LEC 12/16/74 ORIGINAL CODE
E H SCHLOSSER LEC 06/30/78 DELETE RETN K & ADD OPRPIC
E H SCHLOSSER LEC 03/13/79 'DEN'/RADE/CLA SPECIFICATIONS
E H SCHLOSSER LEC 12/26/79 CALL CALCHA & CHK FOR OLD KLSTYP
E H SCHLOSSER LEMSCO 09/16/80 MULTI-CHANNEL RAD DETECTION FILES
E H SCHLOSSER LEMSCO 09/27/80 PRINT CLASS HOG 'AFTER' OPHGN

METHOD

CHECK/CALIBRATE SPECS. OPEN DETECTION FILE. THEN NAME CLADE3/4/5 TO GENERATE DETECTION FILE.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSHK 0 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN 0 GET INTEGER DATA FIELD FROM UNIT 5
MOWARN 0 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
NVIATO 0 NAME 'VIA' 'TO' SUBROUTINES
OPCLA 0 OPEN ALTERNATE PRINT FILE FOR CLASSIFY
SLHCLA 0 'PLOT' (ON PRINTER) SPECTRAL LIMITS FOR CLASSIFY
CALSPA 0 CALIBRATE TRANSFORMATION COEFFICIENTS FOR SPAC140
CALWIN 0 CALIBRATE WINDOW ENVELOPES
CALCHA 0 CALIBRATE CHANNEL POINTERS
HUNIT 0 WRITE HEADING LINE(S) AT TOP OF NEXT PAGE
CLSHOD 0 WRITE COMMON CLASSIFICATION HEADING
OPNOEN 0 OPEN OUTPUT DETECTION FILE (UNIT 21, 22, 23, OR 24) VIA TO EXTERNAL CLA000.
EXTERNAL CLA045.
EXTERNAL CLA345.
CLADE3.CLADE4.CLADES

EXCEPTIONS

1. 'DETECT' MAY NOT BE A DEFAULT COMMAND.
2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE DETECTION FILE.
3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS

<table>
<thead>
<tr>
<th>CONDITION</th>
<th>ACTION</th>
<th>DIAGNOSTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESSING DEFAULT COMMANDS</td>
<td>None</td>
<td>WARNING</td>
</tr>
<tr>
<td>(NWNDW=0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLSTYP SPECIFICATION MISSING</td>
<td>KLSTYP='DEN'</td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>KLSTYP SPECIFICATION INVALID</td>
<td>None</td>
<td>WARNING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOUTCH SPECIFICATION MISSING</td>
<td>NOUTCH=I</td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOUTCH SPECIFICATION &lt; 1</td>
<td>None</td>
<td>WARNING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NOUTCH SPECIFICATION &gt; NLINC</td>
<td>None</td>
<td>WARNING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXTRA SPECIFICATION</td>
<td>None</td>
<td>WARNING</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DATA/CHECKOUT NODE</td>
<td>'VIA' ROUTINE IS CLA000</td>
<td>NONE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WARNING(S) OR FATAL ERROR(S)</td>
<td>WARNING(S) OR FATAL ERROR(S)</td>
<td></td>
</tr>
</tbody>
</table>

GLOBAL DECLARATIONS

INCLUDE KOMXQT.LIST
INCLUDE KOMNER.LIST
INCLUDE KOMKLS.LIST
INCLUDE KOMDEF.LIST
INCLUDE KOMWIN.LIST
INCLUDE KOMWIN.LIST
INCLUDE KOMWIN.LIST
INCLUDE KOMWIN.LIST
INCLUDE KOMWIN.LIST
INCLUDE KOMWIN.LIST
INCLUDE KOMWIN.LIST
INCLUDE KOMWIN.LIST
INCLUDE NULCST.LIST

LOCAL DECLARATIONS

INTEGER INTEMP 8 TEMPORARY

PROCEDURE

CALL TRACE

GET/CHECK TYPE OF FIRST OUTPUT DETECTION CHANNEL

IF(NWNDW.EQ.0) CALL WARN91 'INVALID DEFAULT COMMAND'

IF(KLSTYP.EQ.'DEN') 8 DEFAULT IS DETECTION OF BINARY CLASS DENSITY

CALL GETSKH(KLSTYP(31, NULCST)

IF(KLSTYP.NE.'RAD') GO TO 240

CALL NVIATO(CLASS4S,CLADES) 8 RADIANCE -- NEXT CALL IS TO CLADES

GO TO 300

240 IF(KLSTYP.NE.'DEN') GO TO 250

CALL NVIATO(CLASS4S,CLADE4) 8 DENSITY -- NEXT CALL IS TO CLADE4

GO TO 300

250 IF(KLSTYP.NE.'CLA') GO TO 280

CALL NVIATO(CLASS4S,CLADES) 8 CLASS -- NEXT CALL IS TO CLADES

GO TO 300

280 CALL WARN91 'BAD DETECT TYPE ---'

L-216
CLAOCT 003

MAIN PROGRAMS/ROUTINES

GET/CHK NUMBER OF OUTPUT DETECTION CHANNELS & DRAIN SPECS FOR CURRENT COMMAND

300 NOUTCH=1
   CALL GETSIN(NOUTCH).
   IF(NOUTCH LE 1) GOTO 300
   I.NLIMCM.
   IF((KLSTYP.EQ.'RAD').AND.
      (NOUTCH.LE.1)) CALL MDWARN.
      'MULTI-CHANNEL OEM/CLA DETECTION FILE NOT IMPLEMENTED')
      CALL GETSIN(INTEMP. +1.1.'EXTRA DETECT SPECIFICATION -- ')

CHECK RADIANCE LIMITS

IFILCVLO1.GT.LCVHI11 CALL MDWARN 'NO RADIANCE LIMITS')
IF(IMSATAC.LE.0) GO TO 900 'DATA/CHECKOUT MODE

OPEN PRINT FILE IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER

IF(NWNDOM.LE.0) CALL OPRCLA & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
               NWNDOM=ABS(NWNDOM) NPAGE=0

PLOT SPECTRAL LIMITS & CALIBRATE WINDOW

CALL SLMLCA
   CALL CALSPA
   CALL CALWIN 2.1
   CALL CALCHA

ELIMINATE ANY PART OF NSA OUTPUT WINDOW OUTSIDE OF NSA INPUT WINDOW

MSAOWW(WLIN.WMIN)+MAX(MSAOWW(WLIN.WMIN),MSAOWW(WLIN.WMIN))
MSAOWW(WLIN.WMAX)=MIN(MSAOWW(WLIN.WMAX),MSAOWW(WLIN.WMAX))
MSAOWW(WSAM.WMIN)+MAX(MSAOWW(WSAM.WMIN),MSAOWW(WSAM.WMIN))
MSAOWW(WSAM.WMAX)=MIN(MSAOWW(WSAM.WMAX),MSAOWW(WSAM.WMAX))

ANY NSA DATA AVAILABLE FOR OUTPUT WINDOW??

IF(MSAOWW(WLIN.WMIN).GE.MSAOWW(WLIN.WMIN)) OR.
   IF(MSAOWW(WSAM.WMIN).GE.MSAOWW(WSAM.WMIN)) KLSTYP= 'NUL'
   IF(KLSTYP.EQ.'NUL') CALL MDWARN
   'NO SCAN DATA WITHIN WINDOW')

OPEN OUTPUT DETECTION FILE

IF(INDTOTL.LE.0) GO TO 900
   CALL OPHOEN & OPEN NEW OUTPUT DETECTION FILE OR CHECK EXISTING ONE
   IF(KLSTYP.EQ.'OLD') & IF DESIRED DETECTION DATA ALREADY IN 'OLD' FILE
CALL NVIATO(CLA000,NULSUB) 8 ... THEN ACCEPT IT!!
IF(NALT.GT.0) CALL NDUJ()).(9,10)
IF(NALT.GT.0) CALL CLS000(18)

C ANY DIAGNOSTICS??
C
900 IF(NOTOTL.EQ.0) GO TO 990 8 FORCE AHEAD!
CALL NVIATO(CLA000,NULSUB)
IF(NOPATL.EQ.0) GO TO 920
CALL MDNOTE:
*FATAL ERRORS -- NO DETECTION FILE GENERATED"
GO TO 990
920 IF(NOWARN.EQ.0) GO TO 990
CALL MDNOTE:
*PREVIOUS WARNINGS -- NO DETECTION FILE GENERATED"
IF(NBATCH.EQ.01 WRITE(6,925)
925 FORMAT(4x,"**TRY AGAIN!")

C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KMD= N
CALL MDCLRML NULCS1)
RETURN
END
SUBROUTINE CLADES 0 GENERATE RADIANCE DETECTION FILE

HISTORY

E. H. SCHLOSSER   LEC       03/15/76 ORIGINAL CODE WITH 'CHR' BINS
E. H. SCHLOSSER   LEC       12/20/76 PDBDEF FORMAT BUFFER W/ 'BYT' BINS
E. H. SCHLOSSER   LEC       08/18/80 MULTI-CHANNEL RAO DETECTION FILES

METHOD

READ MSA PIXEL RADIANCE DATA, SCREEN EACH PIXEL AGAINST SPECTRAL LIMITS FOR CURRENT CHANNEL(S), AND POST RADIANCE FOR PIXELS WHICH PASS TO DETECTION FILE.

MACHINE-DEPENDENT CODE

ASSUMES 4 BYTES PER INTEGER.
ASSUMES 20 UNIVAC WORDS PER UNIVAC FASTRAND-FORMATTED DISK SECTOR.

EXTERNAL REFERENCES

MOVIST  MOVE INTEGER STRING
GETRAD  GET RADIANCE IN ALL SELECTED RAW/TRANSFORMED CHANNELS
MOVLOG  PRINT/LOG/COUNT 'NOTE' DIAGNOSTIC MESSAGE
MFATAL  PRINT/LOG/COUNT 'FATAL ERROR' DIAGNOSTIC MESSAGE
PXDMP   DUMP PREAMBLE OF PIXEL BUFFER
GETBYT  GET NON-NEGATIVE NUMBER FROM BYTE STRING
PUTBYT  PUT NON-NEGATIVE NUMBER INTO BYTE STRING
ERIO    INITIATE I/O
ERWAIT  WAIT FOR COMPLETION OF PREVIOUSLY INITIATED I/O
NVIATO  NAME 'VIA' TO SUBROUTINES
        INTEGER NVVBWI  NUMBER OF BYTES FOR NUMBER OF INTEGERS
DOUBLE PRECISION CBPSCS  VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH
DOUBLE PRECISION CBPSIN  VARIABLE-LENGTH CHAR STRING FOR INTEGER
        TO
EXTERNAL CLA000. NULSUB

EXCEPTIONS

1. IF THE NUMBER OF OUTPUT DETECTION CHANNELS REQUESTED BY NOUTCH IS GREATER THAN THE NUMBER OF DETECTION BUFFERS AVAILABLE, THEN A 'WARNING' IS GENERATED AND THE ROUTINE RETURNS.

2. 'BADF' OR 'OFL' ON READING INPUT DATA GENERATES A 'FATAL ERROR' AND
THE ROUTINE RETURNS.

GLOBAL DECLARATIONS

INCLUDE KONKOT.LIST 0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONNNR.LIST 0 COMMON ERTS SCENE PARAMETERS
INCLUDE KONKLTS.LIST 0 COMMON CLASSIFICATION INFO
INCLUDE KONF1T.LIST 0 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE MINDEF.LIST 0 DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KONHLN.LIST 0 COMMON INPUT WINDOW PACKETS
INCLUDE KONJMLN.LIST 0 COMMON OUTPUT WINDOW PACKETS
INCLUDE KONJMT.LIST 0 COMMON DETECTION FILE WINDOWS
INCLUDE KONJML.LIST 0 COMMON 1/O PACKETS FOR DETECTION FILES (E1-E4)
INCLUDE KONJNO.LIST 0 COMMON 1/O FUNCTIONS
INCLUDE KONJLML.LIST 0 COMMON CLASSIFICATION SPECTRAL LIMITS
INCLUDE PX90DEF.LIST 0 DEFINE PIXEL BUFFER STRUCTURE
INCLUDE MAXINT.LIST 0 DEFINE MAXIMUM INTEGER

LOCAL DECLARATIONS

PARAMETER NMBSURS • M • Or NSA PIXEL BUFFERS IN ARRAY
INTEGER INSNS PREAMBLE • (0 NBS • 0 + 0) + (0 EXTRAS * NBS • 0)
PARAMETER NMBSUF • (0 PX9NS-1) • (0 NBS • 0 + 0) + (0 EXTRAS * NBS • 0)
INTEGER NMBSUF(MMBSUF,NMBSURS) 8 ARRAY OF NSA PIXEL BUFFERS
INTEGER IIINBM(NMBSURS) 8 BIN # OF SAMPLE # FOR EACH NSA BUFFER

PARAMETER NOBSURS • M • Or DETECTION PIXEL BUFFERS IN ARRAY
INTEGER INTS PREAMBLE • (0 NBS • 0 + 0) + (0 EXTRAS * NBS • 0)
PARAMETER IDETBUF • (0 PX9NS-1) • (0 NBS • 0 + 0) + (0 EXTRAS * NBS • 0)
INTEGER IDETBURF(IDETBUF,NOBSURS) 8 ARRAY OF DETECTION PIXEL BUFFERS
INTEGER IDEREC • 8 INTEGERS IN DETECTION RECORD

INTEGER IDEBF(PX9NS) 8 GENERAL SHORT DETECTION BUFFER
INTEGER NOUTC • 8 OUTPUT CHANNEL NUMBER (1 TO NOUTC)
INTEGER NBURF • 8 BUFFER NUMBER
INTEGER NDREC • 8 DETECTION RECORD NUMBER
INTEGER NSALIN • 8 CURRENT NSA LINE NUMBER
INTEGER NSALIN,NSALMX • 8 MINIMUM, MAXIMUM NSA LINE IN OUTPUT WINDOW
INTEGER NSAM • 8 CURRENT NSA SAMPLE NUMBER
INTEGER NSAMN,NSAMMX • 8 MINIMUM, MAXIMUM NSA SAMPLE IN OUTPUT WINDOW
INTEGER NSBLO,NSASHI • 8 LOW, HIGH DEFINED NSA SAMPLE IN CURRENT LINE

INTEGER IDEBIF,IDEBSNI • 8 CURRENT, HIGH DETECTION BIN NUMBER

PARAMETER NOINRA=254 • 8 RADIANCE DETECTION PIXEL 'NO INFO' FLAG
PARAMETER NOINRA=255 • 8 RADIANCE DETECTION PIXEL 'NO DATA' THRESHOLD

INTEGER IOSIC • 8 I/O STATUS CODE
PROCEDURE

CALL TRACE

CALL TRACF

CALL INITIALIZC MINIMUM/MAXIMUM SCAN LINES/SAMPLES & DETECTION RECORD LENGTH

IF (NOUTCH.GT.4.NDIBUFFS)  CALL HDWARNI
+ "MORE OUTPUT DETECTION CHANNELS THAN BUFFERS"
IF (NOUTCH.GT.4)  CALL HDWARNI
+ "MORE OUTPUT DETECTION CHANNELS THAN PACKETS"
IF (NDTOL.LT.0.0)  GO TO 900
MSALMN=MSADWM(MLIN.WMIN.NCCT)
MSALRX=MSADWM(MLIN.WMAX.NCCT)
MSASN=MSADWM(MSAM.WMIN.NCCT)
MSASNX=MSADWN(MSAM.WMAX.NCCT)

C

C INTEGERS IN DET REC = (INTS PREAMBLE * (OBSINS+3)/4)
IIIDREC = (PXBINS-1) + (MSASNX-MSASN+3)/4

IF (IIDREC.GT.11DBUFF)  CALL MDFATL('IIDREC > 11DBUFF IN CLADE3')
LOETRS(NCCT)= (IIDREC+87)/28 & CONVERT DETECT REC LENGTH TO UNIVAC SECTORS

C

C INITIALIZE CHANNEL SWITCHES

NLINCM=4
ASSIGN 320 TO LC25W
IF (NLINCM.LT.9)  GO TO 130
ASSIGN 340 TO LC25W
NLINCM=1
GO TO 180

130 ASSIGN 330 TO LC35W
IF (NLINCM.LT.9)  GO TO 140
ASSIGN 360 TO LC35W
NLINCM=2
GO TO 180

140 ASSIGN 340 TO LC45W
IF (NLINCM.LT.9)  GO TO 150
ASSIGN 360 TO LC45W
NLINCM=3

180 CONTINUE
180 CONTINUE

C

C INITIALIZE PREAMBLE IN GENERAL SHORT DETECTION BUFFER

IDENT(PXRECN)=0
IDENT(PXLIQN)=0
IDENT(PXCHAN)=1
IDENT(PXQUAL)=0
IDENT(PXINT)= '0'Y
IDENT(PXBIN)=2
IDENT(PXBSMN)=0

L-221
INITIALIZE BINS IN GENERAL SHORT DETECTION BUFFER TO 'NO INFO' FLAGS

DO 170 IDEBIN=1,IDEBIN
CALL PUSBT(IDET(PKNSIN),IDEBIN,NOINRA)
170 CONTINUE

READ MSA. CLASSIFY. WRITE TO DETECTION FILE FROM LINE MSALMN TO MSALMX

DO 700 MSALIN=MSALMN,MSALMX

-- READ INPUT MSA DATA

CALL GETRAD(MSABUF,(HMBSF),(HMBSF),(OSTC).
   MSALIN,HSASBM,HSASHX)
   IF((OSTC.NE.'BADF') .AND.
   .(OSTC.NE.'OFL')) GO TO 210
   CALL MDFATL(
   .COB408(STC,1,4)) WHILE READING ON UNIT 3'1
   GO TO 800
210 CONTINUE

-- SECURITY BLANKET -- PRINT LINE NUMBER EVERY 90 LINES

IF(MOD(MSALIN,90).NE.0) GO TO 220
   CALL MNOTE('LINE',CHSIN(MSALIN,911)
   IF(MRACE.NE.0) CALL PSTDMP(MSABUF,1,1)
220 CONTINUE

-- COMPUTE ACTUAL LO/MI SAMPLES READ AND BINS WITH SAMPLE 0

MSASLO=MAXINT
MSASHI=MAXINT
DO 230 NBUN=1,NLUNCH
   NSIN(NBUS)=MSASBM(PXLSAM,NBUN)-MSABUF(PXLSAM,NBUN)
   NSIN(NBUS)=MSASLO(PXLSAM,NBUN)-MSABUF(PXLSAM,NBUN)
   NSIN(NBUS)=NSIN(NBUS)-NSIN(NBUS)
230 CONTINUE

-- AFTER QUEUED I/O IS COMPLETE. CLEAR DET BUFFER(S) & INITIALIZE PREAMBLE(S)

DO 280 NOUTC=1,NOUTCH
CALL MAIT1010GTC, LENPRT1, NOUTC)
IF(OSTC.NE.'*') CALL MTFLATL
C000401(OSTC.1.0), WHILE A: NO DETECTION RECORD)
CALL N0V1ST1(IDETF1. NOUTC), IF(REC).
CALL 10ET(11, IPK1INS-1). IDET(IPKINS)

NOREC=NSALIN-NHEADIN(NLIN. MIN. NECT). NOUTC-NOUTC
IDET(IPK1NCM. NOUTC)=NOREC 0 RECORD NUMBER
IDET(IPK2NCM. NOUTC)=NSALIN 0 LINE NUMBER
IDET(IPK3NCM. NOUTC)=NOUTC. O OUTPUT CHANNEL NUMBER
IDET(IPK4NCM. NOUTC)=NSALIN(IPKQUAL. 1)
IDET(IPK5NCM. NOUTC)=NSALIN
IDET(IPK6INC. NOUTC)=NSALIN
IDET(IPK7INC. NOUTC)-IDET(IPK8INC. NOUTC)-MSALIN-MSALE

200 CONTINUE
IF(N0T0TL.NE.0) GO TO 900

C
C -- SCREEN DATA IN BINS OF NSA BUFFER(S) TO FILL BINS OF DET BUFFER(S)
C
:DLBIN[IDETBP(XBIN1.11.1.1)
GO NSALIN-NSALIN-NSALIN
IDBIN=ID0BIN=1
CALL GETBYT('IND0AL).
- MSALIN=MSALIN+MSALIN(1)
 IF(IND0AL.LT.0).OR.(IND0AL.GT.1271) GO TO 900
 GO TO LC3SWH 0 320 (MORE CHANNELS) OR 360 (LAST CHANNEL)

320 CALL GETBYT('IND0AL).
- MSALIN=MSALIN+MSALIN(2)
 IF(IND0AL.LT.0).OR.(IND0AL.GT.1271) GO TO 900
 GO TO LC3SWH 0 330 (MORE CHANNELS) OR 360 (LAST CHANNEL)

330 CALL GETBYT('IND0AL).
- MSALIN=MSALIN+MSALIN(3)
 IF(IND0AL.LT.0).OR.(IND0AL.GT.1271) GO TO 900
 GO TO LC3SWH 0 340 (MORE CHANNELS) OR 360 (LAST CHANNEL)

340 CALL GETBYT('IND0AL).
- MSALIN=MSALIN+MSALIN(4)
 IF(IND0AL.LT.0).OR.(IND0AL.GT.1271) GO TO 900
 IF(IND0AL.LT.0).OR.(IND0AL.GT.1271) GO TO 900

350 CONTINUE
360 CALL PUTBYT('IPK1INS.11.ID0BIN).
- MINDP(IND0AL.1271)
 IF(N0UTCH.11.1) GO TO 980
 CALL PUTBYT('IPK1INS.11.ID0BIN).
- MINDP(IND0AL.1271)
 IF(N0UTCH.11.2) GO TO 980
 CALL PUTBYT('IPK1INS.11.ID0BIN).
- MINDP(IND0AL.1271)
 IF(N0UTCH.11.3) GO TO 980
 CALL PUTBYT('IPK1INS.11.ID0BIN).
- MINDP(IND0AL.1271)

480 CONTINUE

C
C

L-223
C -- QUEUE DETECTION RECORD OUTPUT BUFFER(S) FOR I/O
C
DO 600 NOUTC=1,NOUTCN
   CALL QWDRS(L2NPKT(I,NOUTC), I辩论(I,NOUTC))
600 CONTINUE
C
C LOOP TO READ HSA DATA FOR .XT HSA LINE
C
700 CONTINUE
C
C CLASSIFICATION COMPLETE -- WAIT FOR COMPLETION OF QUEUED I/O
C
DO 700 NOUTC=1,NOUTCN
   CALL WAIT10(IOSTC.L2NPKT(I,NOUTC))
   IF(IOSTC.NE.0) CALL MDFAT1("C8S4CS(IOSTC.I).4) WHILE WRITING DETECTION RECORD")
700 CONTINUE
C
C RETURN TO CONVERSATIONAL MONITOR
C
900 CALL NVIATO(CLA000.NULSUB) a PREVENT CONSECUTIVE CALLS TO CLADE3
RETURN
C
C
C INTERNAL
SUBROUTINE QWDRS; A QUEUE DET RECORD BUFFER FOR WRITING TO DET FILE
C
INTERNAL SUBROUTINE QWDRS: A QUEUE DET RECORD BUFFER FOR WRITING TO DET FILE
C
C
INT 1 XDPKT; A DETECTION PIXEL FILE I/O PACKET
C
INT XDBUF; A DETECTION PIXEL BUFFER
C
C
INTEGER 1 XDPKT(I) a ARGUMENT
C
INTEGER 1 XDBUF(I) a ARGUMENT
C
C
I O SIZE(I XDP KT)=11 DREC
I O SECTION(I XDP KT)=10-((1 XDBUF(PXREC)-1)*LOETRS(NCC) 8 RECN = 1 IN SECTOR 10
I O ADDR(I XDP KT)=LOC(1 XDBUF)
I O FUNC(I XDP KT)="WC" a WRITE
IF(MTRACE.EQ.0) GO TO 400
IF(MOD(I XDBUF(PXLIN)) .501.EQ.0) CALL PXBDMP(I XDBUF)
400 CALL ER10(I XDP KT)
900 RETURN
C
C
C
C
L-224
INTERNAL SUBROUTINE WAITIO  
       0 WAIT FOR COMPLETION OF QUEUED I/O
       0 IOSTC.  0 I/O STATUS CODE (SEE KBMIO FOR CODES)
       0 IOQPKT1 0 I/O PACKET FOR QUEUED I/O

C

INTEGER IOQPKT1 0 ARGUMENT

C

CALL ERMAIT(IOQPKT)
IOSTC=IOCODE(IOQPKT)
RETURN

C

END
SUBROUTINE CLADE4  & GENERATE DENSITY DETECTION FILE

HISTORY
-------

E H SCHLOSSER LEC 01/06/73  ORIGINAL CODE
E H SCHLOSSER LEC 10/28/78  CHANGE NAME FROM CLADE3 TO CLADE4
E H SCHLOSSER LEC 03/13/79  CLASSIFY & ENUMERATE NEAR HITS
E H SCHLOSSER LEC 12/28/79  PX80EF FORMAT BUFFER w/ 'CHR' BINS

METHOD
-------

READ MLA PIXEL RADIANCE DATA. SCREEN EACH PIXEL AGAINST SPECTRAL LIMITS FOR CURRENT CHANNEL(S). ENUMERATE BINARY CLASS DENSITY FOR EACH PIXEL, AND POST DENSITY TO DETECTION FILE.

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

ASSUMES 4 BYTES PER INTEGER.
ASSUMES 8 CHARACTERS PER INTEGER.
ASSUMES 20 UNIVAC WORDS PER UNIVAC FASTRAND-FORMATTED DISK SECTOR.

EXTERNAL REFERENCES
-------------------

MOVIST  & MOVE INTEGER STRING
OETRAD  & GET RADIANCE IN ALL SELECTED RAW/TRANSFORMED CHANNELS
MODATE  & PRINT/LOG/COUNT 'NOTE' DIAGNOSTIC MESSAGE
MDFATL  & PRINT/LOG/COUNT 'FATAL ERROR' DIAGNOSTIC MESSAGE
GETBYT  & GET NON-NEGATIVE INTEGER FROM BYTE STRING
PUTICE  & PUT NON-NEGATIVE INTEGER INTO CHARACTER STRING
GETICE  & GET NON-NEGATIVE INTEGER FROM CHARACTER STRING
ERIO  & INITIATE I/O
ERWAIT  & WAiT FOR COMPLETION OF PREVIOUSLY INITIATED I/O
NVIATO  & NAME 'VIA' 'TO' SUBROUTINES
INTEGER NC4Ni  & NUMBER OF CHARACTERS FOR NUMBER OF INTEGERS
DOUBLE PRECISION C884CS  & VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH NUMBER
DOUBLE PRECISION C884IN  & VARIABLE-LENGTH CHAR STRING FOR INTEGER

EXTERNAL CLA000. NULSUB

EXCEPTIONS
-----------

1. TAPE READ PARITY ERROR. POSITION 0000 GENERATES A NON-FATAL DIAGNOSTIC.
2. ANY OTHER TAPE READ ERROR GENERATES A FATAL DIAGNOSTIC AND TERMINATES.
THE SUBROUTINE.

GLOBAL DECLARATIONS

PARAMETER NMBUFS = 5  # OF MSA PIXEL BUFFERS IN ARRAY
PARAMETER NDBUFS = 9  # OF DETECTION PIXEL BUFFERS IN ARRAY
PARAMETER NBUF = 8  BUFFER NUMBER
PARAMETER NIDET = 8  GENERAL SHORT DETECTION BUFFER
PARAMETER NOINDE = 00  # DENSITY DETECTION PIXEL ‘NO INFO’ FLAG
PARAMETER NODEAD = 20  # DENSITY DETECTION PIXEL ‘NO DATA’ THRESHOLD
PARAMETER INSTAT = 8  GETRAD INPUT STATUS CODE

LOCAL DECLARATIONS

INTEGER MSA BUF = #INTS PREAMBLE + (#BINS+3)/4 + (#EXTRA BYTES+3)/4
PARAMETER NMBUF = (PXBINS-1) * (3548+3)/4 + (19+3)/4
INTEGER MSABUF(NMBUF,NMBUFS)  # ARRAY OF MSA PIXEL BUFFERS
INTEGER NOINSO(NMBUFS)  # BIN # OF SAMPLE 0 FOR EACH MSA BUFFER
PARAMETER NDMBUF = (PXBINS-1) * (3548+2+51)/6 + 1
INTEGER IOETBF(NDMBUF,NDBUFS)  # ARRAY OF DETECTION PIXEL BUFFERS
INTEGER IOET(REC)  # INTEGERS IN 1 DETECTION RECORD
INTEGER INSTAT  # 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 MSASMN, MSASMX  # 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
INTEGER LOCDBS  # LOC IN DETECTION BUFFER OF CURRENT DETECTION BIN + 5
PARAMETER NOINDE = 00  # DENSITY DETECTION PIXEL ‘NO INFO’ FLAG
PARAMETER NODEAD = 20  # DENSITY DETECTION PIXEL ‘NO DATA’ THRESHOLD
INTEGER INSTAT  # GETRAD INPUT STATUS CODE
INTEGER 100000 /*88888*/
INTEGER 110000 /*88888*/
INTEGER 111000 /*88888*/
INTEGER 011100 /*88888*/
INTEGER 001110 /*88888*/
INTEGER 000111 /*88888*/
INTEGER 000011 /*88888*/
INTEGER 000001 /*88888*/

C
C PROCEDURE
C ---------
C
CALL TRACE
C
C INITIALIZE MINIMUM/MAXIMUM SCAN LINES/SAMPLES & DETECTION RECORD LENGTH
C
MSALMN=MSADWN(WLIN,WMIN,NCCT)
MSALMX=MSADWN(WLIN,WMAX,NCCT)
MSASMN=MSADWN(MSAM,WMIN,NCCT)
MSASMX=MSADWN(MSAM,WMAX,NCCT)
MSALIN=MSALMN+1
C
C INTEGERS IN DET REC = #INTS PREAMBLE + (#INS*5)/8
IDREC = (PXBSNS-1) + (MSASMX-MSASMN+3*5)/8
C
IF(IDREC>11DBUF) CALL HDFATL('IDREC > 11DBUF IN CLADE')
LOETR(NCCT)=(IDREC*27)/28 & CONVERT DETECT REC LENGTH TO UNIVAC SECTORS
C
C INITIALIZE CHANNEL SWITCHES
C
NLINCH=4
ASSIGN 320 TO LC2SWH
IF(LINCH(2).LT.9) GO TO 130
ASSIGN 360 TO LC2SWH
NLINCH=1
GO TO 180
130 ASSIGN 330 TO LC3SWH
ASSIGN 430 TO LC3SWN
IF(LINCH(3).LT.9) GO TO 140
ASSIGN 360 TO LC3SWH
ASSIGN 460 TO LC3SWN
NLINCH=2
GO TO 180
140 ASSIGN 340 TO LC4SWH
ASSIGN 440 TO LC4SWN
IF(LINCH(4).LT.9) GO TO 150
ASSIGN 360 TO LC4SWH
ASSIGN 480 to LC45SN
NLINCH=3
150 CONTINUE
160 CONTINUE
C
C INITIALIZE PACKET AND BUFFER ASSIGNMENTS FOR DETECTION FILE
C
CALL PKTOET(LENPKT(INCCT))
1800UT=1
1800=2
180IN=3
180IN=4
NDREC=0
C
C INITIALIZE PREAMBLE IN GENERAL SHORT DETECTION BUFFER
C
1DET(PXRECH)=0
1DET(PXLIN)=0
1DET(PXCAMI)=0
1DET(PXQUAL)=0
1DET(PXBINL)="CHR"
1DET(PXBLIN)=2 & 1DET(PXBLIN)+1 & 1DET(PXHBLIN)+1 USED TO DERIVE JOIN
1DET(PXLSAM)=0
1DET(PXHBLIN)=0
1DET(PXH2AM)=0
1DET(PXLGIN)=NO INDE
1DET(PXNODA)=NO INDE
1DET(PXLJO)=0
1DET(PXHJO)=0
C
C INITIALIZE BINS IN GENERAL SHORT DETECTION BUFFER TO 'NO INFO' FLAGS
C
1DEBHI=NBHIN(1)
DO 170 1DEBHI=1,1DEBHI
     CALL PUTICE(IDET(PXBLINS),(1DEBHI), NO INDE)
170 CONTINUE
C
C CLEAR ALL DETECTION BUFFERS
C
DO 180 1NBUT=1,1NBUTS
     CALL MOVIST(IDETBF(1,NBUF),(1),(1REC).)
          1DEP,(1),(PXBINS-1),(1DEP(PXINS))
180 CONTINUE
C
C READ MSA DATA
C
200 IF(MSALIN.ME.MSALMH) GO TO 500
     MSALIN=MSALIN+1
     CALL GETRAD(MSABUF,(1,NBUF),(NBUF),(INSTAT).
          MSALIN,MSAHIN,MSASM)
     IF((INSTAT.NE."BADF").AND.

L-229
F (INSTAT,NE,"QFL") GO TO 210
CALL MDFATL
  C84C86(INSTAT,1,4)," WHILE READING ON UNIT 3")
      GO TO 900
C
C SECURITY BLANKET
C
210 IF(MOD(MSALIN,50).EQ.0) CALL MNOTE
  = 'LINE',8814(INMSALIN,51)
C 8 IF(MOD(MSALIN,50).EQ.0) CALL PXBDMP(MSABUF(1,1)) & **** TMP DEBUG ****
C
C UPDATE POINTERS & PREAMBLE OF OUTPUT BUFFER
C
IDENTF(PXLINO,18013)=MSALIN ( LINE NUMBER
IDENTF(PXQUAL,18013)=MSABUF(PXQUAL,1)
MSASLO=MAXINT
MSASHI=MAXINT
DO 230 NBUF=1,INLIMCH
      NBUF=MSABUF(PXLBIN,NBUF)-MSABUF(PXLSAM,NBUF)
      MSASLO=MAX0(MSASLO,MSABUF(PXLSAM,NBUF))
      MSASHI=MIN0(MSASHI,MSABUF(PXHSAM,NBUF))
  230 CONTINUE
IDENTF(PXLSAM,18013)=MSASLO
IDENTF(PXHSAM,18013)=MSASHI
IDENTF(PXLBIN,18013)=IDENTF(PXLBIN,18013)+MSASHI-MSASLO
C
C SCREEN DATA IN BINS OF MS BUFFER TO FILL BINS OF DETECTION BUFFER
C
IDENT=IDENTF(PXBIN,18013)-1
LOCBS=NCW41(PXBIN-1)   NEW NUMBER OF CHARACTERS IN BUFFER PREAMBLE
&  IDENT   CURRENT BIN NUMBER
6 59 9 (50 DIVISION BY 6 YIELDS WORD NUMBER)
DO 490 MSASAM=MSASLO,MSASHI
      LOCBS=LOCBS+1
      CALL GETBYF (INOVAL.
      MSABUF(PXBIN,1),MSASAM*NBINS(1))
C
C CLASSIFY PIXEL HI.
C
1 IF(INOVAL.LE.0).OR.(INOVAL.GT.127)) GO TO 490
      GO TO LC3SWH & 320 (MORE CHANNELS) OR 360 (LAST CHANNEL)
   320 CALL GETBYF (MHRADZ.
      MSABUF(PXBIN,2),MSASAM*NBINS(2))
1 IF(MHRADZ.LT.LIMVAL(INDVAL+1.1)) GO TO 421
1 IF(MHRADZ.GT.LIMVAL(INDVAL+1.2)) GO TO 422
      GO TO LC3SWH & 330 (MORE CHANNELS) OR 360 (LAST CHANNEL)
   330 CALL GETBYF (MHRADZ.
      MSABUF(PXBIN,3),MSASAM*NBINS(3))
1 IF(MHRADZ.LT.LIMVAL(INDVAL+1.3)) GO TO 431
1 IF(MHRADZ.GT.LIMVAL(INDVAL+1.4)) GO TO 432
      GO TO LC4SWH & 340 (MORE CHANNELS) OR 360 (LAST CHANNEL)
   340 CALL GETBYF (MHRADZ.
   L-210
C
C MAIN PROGRAMS/RUTINES
C
C ENUMERATE CLASSIFICATION DENSITY FOR HITS
C
370  NWD=LOCDIS/6
371  NCHAR=MOD(LOCDIS,6)+1
372  GO TO 371,372,373,374,375,376,  NCHAR
373  IFETA(NWD-1,IBDIN)+1DETF(NWD-1,IBDIN)+000001
374  IFETA(NWD-1,IBDIN)+1DETF(NWD-1,IBDIN)+000001
375  IFETA(NWD-1,IBDIN)+1DETF(NWD-1,IBDIN)+000001
376  IFETA(NWD-1,IBDIN)+1DETF(NWD-1,IBDIN)+000001
377  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+111100
378  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+111100
379  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+111100
380  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+001110
381  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+001110
382  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+001110
383  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+000111
384  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+000111
385  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+000111
386  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+000111
387  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+000111
388  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+000111
389  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+000111
390  IFETA(NWD,IBDIN)+1DETF(NWD,IBDIN)+000111
391  IFETA(NWD+1,IBDIN)+1DETF(NWD+1,IBDIN)+100000
392  IFETA(NWD+1,IBDIN)+1DETF(NWD+1,IBDIN)+100000
393  IFETA(NWD+1,IBDIN)+1DETF(NWD+1,IBDIN)+100000
394  IFETA(NWD+1,IBDIN)+1DETF(NWD+1,IBDIN)+100000
395
C
C CLASSIFY PIXEL NEAR HIT
C
421  IFIMRAD2.LT.LINIVAL(INOVAL+1,1)-LCVTOL GO TO 420
422  IFIMRAD2.GT.LINIVAL(INOVAL+1,1)-LCVTOL GO TO 420
423  GO TO LC55WN & 430 (MORE CHANNELS) OR 460 (LAST CHANNEL)
430  CALL GETBXY (HRADS.
431  MSABUF(PXBINS,31,MSASAM+NBIINSO(31)
432  IFIMRAD3.LT.LINIVAL(INOVAL+1,1)-LCVTOL GO TO 430
433  IFIMRAD3.GT.LINIVAL(INOVAL+1,1)-LCVTOL GO TO 430
434  GO TO LC55WN & 440 (MORE CHANNELS) OR 460 (LAST CHANNEL)
440  CALL GETBXY (HRADS.
441  MSABUF(PXBINS,41,MSASAM+NBIINSO(41)

L-231
BAN PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

C ENUMERATE CLASSIFICATION DENSITY FOR NEAR HITS

C

C NWD-LOCD0S/8
   NCHAR=MOD(LOCD0S,8)+1
   GO TO(471,472,473,474,475,476), NCHAR

471 I<ETBF(NWD,18D1N3) = I<ETBF(NWD,18D1N3) + 100000
   GO TO 490

472 I<ETBF(NWD,18D1N3) = I<ETBF(NWD,18D1N3) + 010000
   GO TO 490

473 I<ETBF(NWD,18D1N3) = I<ETBF(NWD,18D1N3) + 001000
   GO TO 490

474 I<ETBF(NWD,18D1N3) = I<ETBF(NWD,18D1N3) + 000100
   GO TO 490

475 I<ETBF(NWD,18D1N3) = I<ETBF(NWD,18D1N3) + 000010
   GO TO 490

476 I<ETBF(NWD,18D1N3) = I<ETBF(NWD,18D1N3) + 000001
   GO TO 500

C

C LOOP BACK TO CLASSIFY NEXT PIXEL IN THIS LINE

490 CONTINUE

C

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

C

C

C QUEUE DETECTION RECORD OUTPUT BUFFER FOR I/O

C

C

C CLASSIFICATION COMPLETE

C

C MSAOWW(WLIN.WMAX.NCCT1=MSAOWIN

900 CALL NVIATO(CLAOGO,NULSUB) & PREVENT CONSECUTIVE CALLS TO CLAOE4
   RETURN

C
SUBROUTINE PKTOET

INTERNAL

INTEGER IXOPKT(1) ; ARGUMENT
INTEGER IXDBUF(1) ; ARGUMENT

100 SIZE(IXOPKT)=100REC
RETURN

ENTRY RITDBF

WRITE RECORD FROM DETECTION BUFFER TO DETECTION FILE

IF (ICODE(IXOPKT).EQ.' 1' GO TO 300
CALL HDFATL
= GBYCS(ICODE(IXOPKT),1,4), ' WHILE WRITING DETECTION RECORD')
GO TO 300
300 NDREC=NDREC+1
IXDBUF(PXRECH)=NDREC
CALL DETICE(IXDBUF(PXIN)),
= IXDBUF(PXIN),(IXDBUF(PXIN)-1)
CALL DETICE(IXDBUF(PXHJ01),
= IXDBUF(PXHJS),(IXDBUF(PXHJS)-1)
10SECT(IXOPKT)=10+(NDREC-1)+LOE(IXCT)
NDREC = 1 GOES IN SECTOR 10
IGADDR(IXOPKT)+LOC(IXDBUF)
98 FUNC(IXOPKT)=9C ' WRITE
C 8 IF (MOD(IXDBUF(PXINO).50).EQ.0) CALL PXBOM(PXDBUF) & TMP DEBUG ****
CALL ERIO(IXOPKT)
900 RETURN

END
SUBROUTINE CLADES & DETECT CLASS (PHASE 5)

C (E H SCHLOSSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

MNOTE
NVIATO

EXTERNAL CLA129, CLADES
CALL TRACE

CALL MNOTE('DETECT,CCLASS NOT YET IMPLEMENTED')
CALL NVIATO(CLA129, PICDE9)
RETURN
END
SUBROUTINE CLASED 9 DETECT NXS-DERIVED DATA (PHASE 9)

C (E H SCHLOSSER)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

included

INCLUDE KOMKOT.LIST
EXTERNAL CLA000,NULSUB
CALL TRACE

C ON RETURN, CALL CLA000 TO GET COMMANDS
CALL NVIATO(CLA000,NULSUB)

C CHECK DIAGONOSTIC COUNTERS

IF(NDMARN.EQ.0) GO TO 820
CALL NDNOTE('PREVIOUS WARNINGS -- NO DETECTION FILE GENERATED')
IF(NDBATCH.EQ.0) WRITE(6,019)
019 FORMAT('...TRY AGAIN')
GO TO 900
820 IF(NDMFATL.EQ.0) GO TO 860
CALL NDNOTE('PREVIOUS FATAL ERRORS -- NO DETECTION FILE GENERATED')
GO TO 900
860 IF(NCHECK.EQ.0) GO TO 890
CALL NDNOTE('CHECKOUT MODE -- NO DETECTION FILE GENERATED')
GO TO 900

C PREPARE FOR NEXT DISPLAY
900 NWNDOW=NWNDOW+1

C CLEAR WARNINGS & RETURN TO NEXT STATEMENT IN CALLING ROUTINE
900 CALL NDCLRM(-0)
RETURN
END
MAIN PROGRAMS/ROUTINES

SUBROUTINE CLAEXI 8 TERMINATION ROUTINE FOR CLASSIFY

HISTORY

E H SCHLOSSER  LEC  12/10/79  ORIGINAL CODE
E H SCHLOSSER  LEC  12/26/79  CALL CLOSPN. NOT CLOSPN 8 CLOSE3

METHOD

CLOSE INPUT & OUTPUT FILES & TERMINATE PROGRAM.

MACHINE-DEPENDENT CODE

NONE.

GLOBAL DECLARATIONS

INCLUDE KONKOL.LIST  8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONKLS.LIST  8 COMMON CLASSIFICATION INFO

LOCAL DECLARATIONS

NONE.

PROCEDURE

CALL TRACE

CLOSE/VERIFY INPUT MS4 FILE (UNIT 3) IF OUTPUT DETECTION FILE HAS GENERATED
WRITE(8,125)
125 FORMAT(*,"**PROGRAM TERMINATION")
IF(MDATAC.NE.0) GO TO 900  8 DATA/CHECKOUT MODE
IF(KLSTYP.EQ."RAD") OR,  8 IF OUTPUT FILE WAS GENERATED ...
  (KLSTYP.EQ."DEN") OR,
  (KLSTYP.EQ."CLA") CALL CLOSE3  8 ... THEN CLOSE INPUT FILE

CLOSE OUTPUT FILES

IF(MHNDOM.GT.0) CALL CLOSPN
CALL CLOSPNR

L-236
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C
C TERMINATE PROGRAM
C
C 900 CALL PSTOP("****PLEASE FREE J. OR DREWIND J. OR DLOCATE J."")
C
C PSTOP DOES NOT RETURN
C
END
ROUTINE CLARADI 8 GET/CHECK RADIANCE LIMITS
JMD1 8 I: FIRST 3 CHARS OF COMMAND 0: SPACES

CALLOSER:

INAL SUBROUTINES/FUNCTIONS CALLED

READS
MOLOG

INCLUDE KOMXGT.LIST  8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST  8 COMMON CLASSIFICATION INFO
INCLUDE KOMSLM.LIST  8 COMMON SPECTRAL LIMITS
CALL TRACE

CHECK POINTERS

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

C
C CHECK INDEX RADIANCE VALUES TO SEE IF CHANNELS HAVE JUST BEEN CHANGED
C
IF(LCVLO1.LE.LCVHII) GO TO 500  8 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
      LIMVAL(INDVAL+1,N) = 99999999  8 MINIMUM
      LIMVAL(INDVAL+1,N) = 99999999  8 MAXIMUM
440 CONTINUE
450 CONTINUE
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 --')
   LCVLO1=MIND(LCVLO1,INDVIN)
   LCVHII=MAX(LCVHII,INDVAX)
   INDVAL=INDVIN

L-238

ORIGINAL PAGE IS
OF POOR QUALITY
C GET/CHECK LIMIT RADIANCE VALUES
C
C DO 550 N=2.NLIMCH
   IF(LIMCH(N).EQ.999) GO TO 552
      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)=MIN0(LCVLO(N).LIMVAL(INDVAL+1.2*N-3))
      LCVHI(N)=MAX0(LCVHI(N).LIMVAL(INDVAL+1.2*N-2))
   550 CONTINUE
      CALL GETSIN(ITEMP, +1,-1, 'EXTRA RADIANCE SPECIFICATION --**
C
C DUPLICATE SAME LIMIT VALUES FOR REMAINING INDEX VALUES
C
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 DONE
C
C 900 KOMD=' RETURN
C
C END
SUBROUTINE CLAXQT  A INITIALIZATION ROUTINE FOR CLASSIFY

-----------------------------

HISTORY
-------

E H SCHLOSSER  LEC  11/05/74  ORIGINAL CODE
E H SCHLOSSER  LEC  01/31/79  ALLOW DEFAULT COMMANDS FROM MACDAM
J C CRISP  LEC  12/16/79  WSP100 & 100 REPLACE WINC & I

METHOD
------

INITIALIZE PROGRAM, OPEN FILES, IDENTIFY SCENE, QUEUE DEFAULT COMMANDS.

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

UNIVAC EXEC-6 PROGRAM FILE NAMING CONVENTIONS.
DIMENSION & FORMAT SPECIFICATIONS ASSUME 6 CHARACTERS PER WORD.

EXTERNAL REFERENCES
-------------------

NVIATO  & NAME NEXT 'VIA' & 'TO' SUBROUTINES
PSTART  & PROGRAM START INITIALIZATION
OPEN3  & OPEN INPUT FILE 3 (INPUT MSS OR RBV DATA)
LDREG8  & LOAD REGISTRATION PARAMETERS FROM FILE 8
IDLU3  & IDENTIFY FILE 3 HARDWARE
IDERTS  & IDENTIFY MSS OR RBV DATA
SYSSID  & ADD DISK SYMBOLIC FILE OR ELT TO SYSIN RUNSTREAM
MDFALT  & SUBMIT FATAL DIAGNOSTIC MESSAGE

EXTERNAL CLAXQT. NULSUB

EXCEPTIONS
----------

1. MISSING DEFAULT COMMANDS GENERATE A FATAL DIAGNOSTIC.

GLOBAL DECLARATIONS
---------------------

INCLUDE KOMXOT.LIST  & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST  & COMMON EATS SCENE PARAMETERS
INCLUDE KOMFIT.LIST  & COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE WINDEF.LIST  & DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMOWN.LIST  & COMMON OUTPUT WINDOW PACKETS

L-240
C LOCAL DECLARATIONS

C ------------------

C INTEGER LOCFILE  -- LOCATION WITHIN DISK SYMBOLIC FILE ( IF > 0 )

C ---

C IDENTIFY PROGRAM

C

CALL PSTART( 'DAM CLASSIFY(8009)' )

C

C OPEN ERTS FILE & LOAD REGISTRATION PARAMETERS

C

CALL OPENS

CALL LDREGB  -- FATAL ERROR IF REGISTRATION PARAMETERS NOT ON UNIT B

C

C IDENTIFY ERTS SCENE

C

IF(MDATA.NE.0) GO TO 300  -- DATA/CHECKOUT MODE

IF(MDATA.EQ.0)

& ENCODE(10,195,JH00) NERTS.NCCT.NCCTOT

185 FORMAT( 'E-'.JI,J4,'--'.JI,'/'.JI)  -- COMPACT TO FIT ON BOX PAGE

WRITE(8,185) NERTS.NCCT.NCCTOT

WRITE(8,195)

195 FORMAT(IX)  -- SKIP LINE

CALL IOLN(6)

CALL IOLRTS(6)

C

C INITIALIZE SPACING

C

MSONW(WLIN.WSP100)=100

MSONW(WSAM.WSP100)=100

C

C QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE

C

320 CALL SYSAOD(LOCFILE, 'MACDAM', 'DEF-CLASSIFY', ' ')

IF(LOCFILE.LE.0) CALL SYSAOD(LOCFILE, 'DAM', 'DEF-CLASSIFY', ' ')

IF(LOCFILE.LE.0) CALL MDFATL('NO DEFAULT COMMANDS')

C

C RETURN, THEN HAVE MAIN CALL CLAO00 TO GET DEFAULT/USER COMMANDS

C

CALL NVIAL(0)  -- CLAO00.NULSUN)

RETURN

END
SUBROUTINE CLS02N  • CLOSE OUTPUT DETECTION FILE

HISTORY

E H SCHLOSSER  LEC  01/05/74  ORIGINAL CODE IN CLS02N
MARY TOMPKINS  LEC  12/13/79  REV HEADER & REPLACE FLO BY CST4IN

METHOD

CHECK KLSTYP:
  IF KLSTYP = 'NUL' THEN NO DETECTION FILE EXISTS (OR SHOULD EXIST) --
      FREE THEN DELETE FILE (IF ANY).
  IF KLSTYP = 'OLD' THEN EXISTING DETECTION FILE IS VALID --
      DO NOT OVER-WRITE ITS HEADER.
  OTHERWISE --
      MOVE KOMNER, KOMKLS, KOMFIT, AND KOMDET INTO BUFFER. WRITE FINAL DETECTION HEADER
      FROM BUFFER. AND CATALOG FILE.

MACHINE-DEPENDENT CODE

ASSUMES 28 UNIVAC WORDS PER UNIVAC FASTRAND-FORMATTED DISK SECTOR.
FORCES SECTOR ALIGNMENT OF EACH COMMON IN THE HEADER.

EXTERNAL REFERENCES

ERCSF     8 SUBMIT EXEC-B CONTROL STATEMENT FUNCTION
ERIOW     8 INITIATE I/O & WAIT FOR COMPLETION
CST4IN     8 CHAR STRING FOR INTEGER

EXCEPTIONS

NONE

GLOBAL DECLARATIONS

INCLUDE KOMXOT.LIST  8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMLEN.LIST  8 COMMON I/O PACKETS FOR DETECTION FILES (21-24)
INCLUDE KOMIO.LIST  8 FORTRAN MANIPULATION OF ASSEMBLER I/O PACKETS
INCLUDE KOMNER.LIST  8 COMMON ERTS SCENE PARAMETERS
INCLUDE KOMKLS.LIST  8 COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST  8 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMDET.LIST  8 COMMON DETECTION FILE WINDOWS
INCLUDE WINDEF.LIST  8 DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE FACBIT.LIST  6 EXEC-0 FACILITY REQUEST STATUS BIT MNEMONICS

LOCAL DECLARATIONS

INTEGER HMD  6 WORD NUMBER IN HEADER BUFFER AND/OR COMMON BLOCK
INTEGER KBUFIR(29,9)  6 INTERNAL WORK BUFFER FOR DETECTION FILE HEADERS

CHAR 0000000001111111111112222222223
& JASOA2(4)'JASO.AX *DAMDET-N. .'/
& JFREE2(4)'JFREE-E .ZN . .'/
& JFRED2(4)'JFREE-D .ZN . .'/

PROCEDURE

CALL TRACE

IS DETECTION FILE NON-EXISTENT. GOOD-OLD, BAD, GOOD-NEW?

IF (NCCT.LT.1) OR (NCCT.GT.4) 00 TO 900
IF(KLSTYP.EQ.'NUL') 00 TO 900  8 FILE DOES NOT (OR SHOULD NOT) EXIST
IF(KLSTYP.EQ.'OLD') 00 TO 900  8 DON'T TOUCH THAT GOOD OL' FILE!!
IF(NDFATL.NE.0) 00 TO 900  8 FILE BAD -- DON'T REPLACE INTERIM HEADER

GOOD NEW FILE -- MOVE KOMNER, KOMKLS, KOMFIT, AND KOMDET INTO BUFFER

DO 811 NMD=1,SIZNER
  811 KBUFIR(NMD,1)=KOMNER(NMD)
DO 812 NMD=1,SIZKLS
  812 KBUFIR(NMD,4)=KOMKLS(NMD)
DO 813 NMD=1,SIZFIT
  813 KBUFIR(NMD,6)=KOMFIT(NMD)
DO 815 NMD=1,SIZDET
  815 KBUFIR(NMD,8)=KOMDET(NMD)

WRITE FINAL HEADERS FROM BUFFER

CALL WRHDR(L2NPKT,1,NCCT)

CATALOG DETECTION FILE

800 CALL CST41N(JFREE2,17,1, NCCT,1)
CALL ERCSF(NAO,JFREE2)
CALL CST41N(JASOA2,17,1, NCCT,1)
CALL ERCSF(NAO,JASOA2)  8 PROTECT UNTIL RUN TERMINATES
IF(KLSTYP.EQ.'NUL') CALL CST41N(JFRED2,17,1. NCCT,1)
IF(KLSTYP.EQ.'NUL') CALL ERCSF(NAO,JFRED2)
DONE

900 RETURN

INTEGER NPKT

IFUNC(NPKT) = 'AC'  & WRITE
IOSIZE(NPKT) = 20+9  & HEADER SIZE IN WORDS
IADDR(NPKT) = LOC(KBUF)
IOSECT(NPKT) = 0
CALL ERIO(NPKT)
IOSIZE(NPKT) = LDETRS(NCCT) + 20  & RECORD SIZE IN WORDS
RETURN

END
SUBROUTINE OPNOR8  a OPEN/CHECK OUTPUT DETECTION FILE

HISTORY

Original Code
E. H. SCHLOSSER  LEC  01/03/74
Mary Tompkins  LEC  12/13/79
Mary Tompkins  LESCO  05/16/80

Method

Attempt to assign new detection file. If file is already assigned
or file is cataloged and not presently in use, read header of
file into buffer. Compare contents of buffer to KOMER,
KOMKL. KOMFIT. AND KOMDET. If comparison is not equal reinitialize
detection file header to binary zeros before returning.

Machine-Dependent Code

One UNIVAC FASTRAND-FORMATTED DISK SECTOR IS 20 UNIVAC 1100 WORDS.

External References

ERCSF  a SUBMIT EXEC-B CONTROL STATEMENT FUNCTION
NONOTE  a PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
MFATL  a PRINT/COUNT/LOG 'FATAL ERROR' DIAGNOSTIC MESSAGE
ERIOR  a INITIATE 1/O & WAIT FOR COMPLETION
CST4IN  a CHAR STRING FOR INTEGER
DOUBLE PRECISION CSS4CS & VARIABLE LENGTH CST FOR FIXED LENGTH CST

Exceptions

1. IF THE DETECTION FILE EXISTS BUT IS BEING USED BY ANOTHER RUN,
   THEN ISSUE 'FATAL ERROR' AND RETURN.

2. IF I/O ERRORS ARE ENCOUNTERED WHILE READING OLD HEADER OR WRITING
   INTERIM BLANK HEADER, THEN ISSUE 'FATAL ERROR' AND RETURN.

Global Declarations

INCLUDE KOMKQT.LIST  a COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMLEN.LIST  a COMMON 1/O PACKETS FOR DETECTION FILES (21-24)
INCLUDE KOMO.LIST  a COMMON 1/O FUNCTION
INCLUDE KOMN.LIST  a COMMON 1100 SCENE PARAMETERS
INCLUDE KOMKLS.LIST
8 COMMON CLASSIFICATION INFO
INCLUDE KOMFIT.LIST
8 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMOMN.LIST
8 COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMDET.LIST
8 COMMON DETECTION FILE WINDOWS
INCLUDE MINDEF.LIST
8 DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE PX8DEF.LIST
8 DEFINE STRUCTURE OF PIXEL BUFFER
INCLUDE FACBIT.LIST
8 EXEC-8 FACILITY REQUEST STATUS BIT Mnemonics

C LOCAL DECLARATIONS
C -----------------
C INTEGER
C CHAR 000000001111111111111111111111111111111111111111111111
C 123456789012345678901234567890123456789012345678901234567
7 JASOCl(8l,'4A90,CP •OANOV-N..rfl/POS/46 . DETECTION FILE '/
4 JASOCl(81,'4AOG-A0 •OAMDE7-N.. /
4 JOUS8f411 00AMOEt-N.	/

C INTEGER KBUFR(28,9) 8 INTERNAL WORK BUFFER FOR DETECTION FILE HEADERS
C INTEGER IROCT 8 COMPARISON ERROR COUNTER
C INTEGER ISTAT 8 I/O STATUS CODE (CST)

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

C INITIALIZE COMPARISON ERROR COUNTER
C IROCT=0

C INITIALIZE DETECTION WINDOW, DATE, AND RECORD LENGTH IN SECTORS
C DO 110 NCT=1,4
MSADNW(MLIN.WMIN.NCT)=0
MSADNW(MLIN.WMAX.NCT)=0
MSADNW(HSAM.WMIN.NCT)=0
MSADNW(HSAM.WMAX.NCT)=0
JENMDY(NCT)=0
LOETRS(NCT)=0
110 CONTINUE
IF((NCT.LT.11).OR.(NCT.GT.4)) GO TO 900
MSADNW(MLIN.WMIN.NCT)=MSAOWW(MLIN.WMIN)
MSADNW(MLIN.WMAX.NCT)=MSADNW(MLIN.WMAX)
MSADNW(HSAM.WMIN.NCT)=MSADNW(HSAM.WMIN)
MSADNW(HSAM.WMAX.NCT)=MSADNW(HSAM.WMAX)
JENMDY(NCT)=JMDY
LOETRS(NCT)=0 8 RECORD LENGTH IN SECTORS NOT DEFINED AT FILE OPEN

C PUT (20+NCT) INTERNAL DETECTION FILE NAME IN ALL LUEN I/O PACKETS
C

L-248
DO 120 NPKT=1,4
   CALL MOVClST(LBNPKT(I,NPKT,(1,1,1,1),CALL PUTICE(LBNPKT(I,NPKT),8)
   ICE=8)+NCCT
120 CONTINUE

C C C TRY TO ASSIGN NEW DETECTION FILE C
C CALL C141N(JASOC2,17,1, NCCT,1)
WRITE(6,129) JASOC2
129 FORMAT(1&('88.881')
CALL ERC SF(NAO,JASOC2)
CALL C141N(JUSEB,7,1, NCCT,1)
CALL C141N(JUSEB,17,1, NCCT,1)
CALL ERC SF(TEMP,JUSEB)
IF(ALREADY(NAO).EQ.0) GO TO 150
CALL MONOTE('DETECTION FILE ALREADY ASSIGNED TO THIS RUN')
GO TO 400
150 IF(PRVCAT(NAO).EQ.0) GO TO 500 0 SUCCESSFULLY ASSIGNED NEW DETECTION FILE
C
C OLD DETECTION FILE EXISTS -- ASSIGN IT
C CALL MONOTE('DETECTION FILE ALREADY CATALOGED')
C CALL C141N(JASOA2,17,1, NCCT,1)
C CALL ERC SF(NAO,JASOA2)
IF(OTHRUN(NAO).EQ.0) GO TO 400
CALL HDFAIL(1, 'DETECTION FILE IN USE BY ANOTHER RUN - RUID/QUAL NOT UNIQUE')
GO TO 900
C
C READ HEADER (SECTORS 0 THRU 8) INTO BUFFER
C 400 CALL RDHDR
C
C COMPARE SECTORS 0 THRU 8 WITH KOMNER, KOMKLS, AND KOMFIT
C DO 441 NWD=1.67
   IF(KOMNER(NWD).NE.KBUF(R(NWD,1))) IROCNT=IROCNT+1
441 CONTINUE
DO 442 NWD=1.27
   IF(KOMKLS(NWD).NE.KBUF(R(NWD,4))) IROCNT=IROCNT+1
442 CONTINUE
DO 443 NWD=1.17
   IF(KOMFIT(NWD).NE.KBUF(R(NWD,6))) IROCNT=IROCNT+1
443 CONTINUE
C
C COMPARE KOMDET WITH SECTOR 7 THRU 8
C DO 444 NWD=1.24 8 DATES NEED NOT MATCH
   IF(KOMDET(NWD).NE.KBUF(R(NWD,8))) IROCNT=IROCNT+1

L-247
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

464 CONTINUE

C CHECK COMPARISON ERROR COUNTER
C
IF(INCMT.NE.0) GO TO 470
CALL MDNOTE( 'EXISTING DETECTION FILE OK -- NOT REPLACED')
KLSTYP = 'OLD'
GO TO 900

470 CALL MDNOTE( 'EXISTING DETECTION FILE REPLACED')

C WRITE INTERIM BLANK HEADER OF BINARY ZEROS IN SECTOR 0
C
500 DO 920 NWD=1,20
KBUFR(NWD,1)=0
920 CONTINUE
CALL WRHDR
C
DONE
C
900 RETURN

C
C INTERNAL
SUBROUTINE WRHDR
C
IOFUNC(LZ1PKT)="8K"  READ
00 TO 200
C
ENTRY WRHDR
C
IOFUNC(LZ1PKT)="8C"  WRITE
C
200 IOSIZE(LZ1PKT)=28*9  HEADER IS 28 WDS/SECTOR TIMES 9 SECTORS
IADDR(LZ1PKT)=LOC(KBUFR)
I0SECT(LZ1PKT)=0
CALL ER:02(LZ1PKT)
I STAT=10CODE(LZ1PKT)
IF(I STAT.NE.' ') CALL MDFATL
* C054CS(I STAT,1,-1) WHILE OPENING DETECTION FILE HEADER
I0SIZE(LZ1PKT)=0  RECORD SIZE IN WORDS NOT DEFINED AT FILE OPEN TIME
RETURN
C
END

L-248
SUBROUTINE OPRCLA 0 OPEN ALTERNATE PRINT FILE FOR CLASSIFY

HISTORY
------
E N SCHLOSSER LEC 06/28/76 ORIGINAL CODE

METHOD
------
A MAXIMUM OF 1 ALTERNATE PRINT FILE IS OPENED & INITIALIZED.

MACHINE-DEPENDENT CODE
-----------------------
NONE.

EXTERNAL REFERENCES
---------------------
OPENPR OPEN ALTERNATE PRINT FILES

EXCEPTIONS
-----------
NONE.

INCLUDE KOMXOT.LIST
CALL TRACE

NO MORE THAN 1 ALTERNATE PRINT FILE FOR CLASSIFY

IF(MALT.LE.0) GO TO 900 8 NO ALTERNATE PRINT FILES
MALT=MIND(MALT,1)

OPEN FILE(S)

IF(MDATA.NE.0) GO TO 900 8 DATA/CHECKOUT MODE
CALL OPENPR

IDENTIFY ERTS SCENE

WRITE(10,225)
225 FORMAT(1X) 8 SKIP LINE
CALL IDLUS(10)
CALL IDENTS(10)
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C
C
900 RETURN
END
SUBROUTINE SLNCLA  PLOT SPECTRAL LIMITS FOR CLASSIFY

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

INCLUDE KOMKGT.LIST
INCLUDE KOMKLS.LIST
INCLUDE KOMSLM.LIST
CALL TRACE

PLOT SPECTRAL LIMITS

IF(MALT,LE.,0) GO TO 900  8 NO ALTERNATE PRINT FILES
LCVL=MAX0((LCVLOI-1),0)
LCVN=MAX0((LCVLOI-90),(LCVNI1+10))
LCVH=MINO(LCVH,127)
LCV10=10*((LCV10+0)/10)
DO 880 N=2,LCVH
CALL LIMPLT(N)
880 CONTINUE
900 RETURN

SUBROUTINE LIMPLT(N)
DIMENSION LSYM(128)
CALL MOUNIT(1,10)
LCVL=MAXO((LCVLOI-2),0)
IF((LCVH(N).GT.299)) CALL MOUNIT("LIMIT CHANNEL VALUE > 127'")
LCVH=MINO((LCVH(N)+2),299)
WRITE(10,300) INT(D/O(NWD),NWD=1.4),LIM(N)
300 FORMAT(1 'SPECIFIC LIMITS FOR ',CH1,'/
2 ',CH1,NL
NLV2=N(N-1)
NLV1=NLV2-1
DO 300 LCV=LCVH,LCVL,-1
DO 300 INDVAL=ICVL,LCVH
IF((LCV.LT.INDVAL+INDVAL+1,NLV1)) GO TO 340
IF((LCV.GT.INDVAL+INDVAL+1,NLV2)) GO TO 340
LSYM(INDVAL+1)="X"
340 CONTINUE
340 LSYM(INDVAL+1)="/

L-251
380 CONTINUE
  IF(MOD(LCV,10).NE.0) GO TO 380
  DO 370 INDVAL=ICVL,ICVM,10
  370 IF(LSYM(INDVAL+1).EQ.' ') LSYM(INDVAL+1)='+'
  380 WRITE(10,385) LCV,(LSYM(INDVAL+1).INDVAL=ICVL,ICVM)
  385 FORMAT(IX,J5,128A1)
  390 CONTINUE

WRITE GRAPHIC CALF FOR INDEX CHANNEL

400 FORMAT(4X,128A1)
  DO 100 INDVAL=ICVL,ICVM
  100 LSYM(INDVAL+1)=INDVAL/100
  WRITE(10,400) (LSYM(INDVAL+1).INDVAL=ICVL,ICVM)
  DO 10 INDVAL=ICVL,ICVM
  10 LSYM(INDVAL+1)=(INDVAL-100*(INDVAL/100))/10
  WRITE(10,400) (LSYM(INDVAL+1).INDVAL=ICVL,ICVM)
  DO 1 INDVAL=ICVL,ICVM
  1 LSYM(INDVAL+1)=(INDVAL-10*(INDVAL/10))
  WRITE(10,400) (LSYM(INDVAL+1).INDVAL=ICVL,ICVM)
  WRITE(10,500) LIMCH1
  500 FORMAT('0 CHANNEL ',I1)
RETURN
END
PROGRAM PRTDET

--------------

HISTORY
--------

E N SCHLOSSER  LEC  07/02/73  ORIGINAL CODE
E N SCHLOSSER  LEMSCO  09/16/80  KONALT/SYN/TBL/KS.PRTDENS TO PRTDET

METHOD
-------

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

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

NONE.

EXTERNAL REFERENCES
-------------------

NVIATO  & NAME 'VIA' 'TO' ROUTINES
VIATO   & CALL 'VIA' 'TO' ROUTINES
VIA     TO EXTERNAL PR0000, PRDXQT

EXCEPTIONS
----------

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

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

GLOBAL DECLARATIONS
-------------------

INCLUDE KOMQOT.LIST  # COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMLOG.LIST  # COMMON LOG FILE BUFFER, I/O PKT. POINTERS
INCLUDE KOMIUS.LIST  # COMMON POINTERS/FLAGS/BUFFER FOR UNIT S
INCLUDE KOMLN2.LIST  # COMMON I/O PKTS FOR DETECTION FILES UNITS 2N
INCLUDE KOMIMN.LIST  # COMMON INPUT WINDOW PACKETS
INCLUDE KOMONW.LIST  # COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMNNR.LIST  # COMMON ERTS SCENE PARAMETERS
INCLUDE KOMLKL.LIST  # COMMON CLASSIFICATION INFO

L-253
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

INCLUDE KONFIT.LIST COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMALT.LIST COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS
INCLUDE KOMSYM.LIST COMMON SYMBOL TABLE
INCLUDE KOMTBL.LIST COMMON MULTI-PURPOSE TABLE
INCLUDE KOMKS.LIST COMMON COLOR SCREEN PARAMETERS
INCLUDE KOMDET.LIST COMMON DETECTION FILE WINDOW PACKETS & DATES

C
C PROCEDURE
C

CALL NVIAA0(PRDOO0,PROXQT) FIRST CALL IS VIA PRDOO0 TO PROXQT
100 CONTINUE
CALL VIATO
GO TO 100
END (STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE)
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 09/15/80 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 FIELD DATA 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.PRDDET-MAP & SYMBOIC 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
  8 = BATCH

L-256
(18XOT OPTIONS ARE PRELOADED BY EXEC INTO REGISTER AS IN MASTER BIT NOTATION.

LOCAL DECLARATIONS

AXRS
$1(00) . D-BANK
SSSH FORM 6.6.18
LABSDF $SSH 050.1. 'F' .0 . LABEL. 1 WD. FORTRAN. FIELDATA
LABING "SOFF".
MAPSDF $SSH 000.9.0.0 . DATA. 9 WDS. . FIELDATA
MAPING '8XQT: MAP. FZN DAM.PRTDET-MAP. PRTDET . :8XQT'
ADDSDF $SSH 000.9.0.0
ADDING '8XQT: ADD. DAM.SYS-MAPOPT . :8XQT'
XOTING $SSH 000.9.0.0
XOTING '8XQT: XQT.1 PRTDET . :8XQT'
EOFSDF - 0 . END-OF-FILE STOP WORD
PF FORM 12.6.17
CSFASO '8ASO. T 20. '
CSFADD '8ADD 20. '
SAVREG RES 1
IOPKT 130D '20'. MS 33. LABSDF. '0' 0

PROCEDURE

$1(01) . I-BANK
PRTDENS LAU 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. (CSFASO) . ADDRESS OF 8ASO IMAGE
ER CSFS . DO IT
SA A0. SAVREG . STORE &
PRINT (PF 2.1. SAVREG) . PRINT 8ASO STATUS
GETOPT . LOAD OPT LTRS INTO A2.A3.A4
PUTOPT DS A2. XOTING+2 . STORE OPTION LETTERS INTO 2XQT IMAGE
SA A4. XOTING+4 . (3 WARDS -- MAX 18 OPT LETTERS)
WRITE LA A0. (IOPKT) . ADDRESS OF I/O PACKET
ER 10WS . WRITE SDT IMAGES TO 20.
ADD LA A0. (CSFADD) . ADDRESS OF SADD IMAGE
ER CSFS . DO IT
ER EXITS .
END PRTDENS

L-257
PRTDET OVERLAY STRUCTURE

HISTORY

E H SCHLOSSER LEC 08/19/74 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 01/06/80 PEEK.POEK.IF.FI & PHASE 3.4.5 SEO
E H SCHLOSSER LEMSCO 05/16/80 CHA.COL.INT.PIC.TAB. PRTDENS TO DET

LIB DAM.

SE0 S-MAIN
IN DAM.PRTDET/ 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

SE0 S-PRO0129+
IN DAM.PRD000 CALL USER-SPECIFIED PHASE 0 ROUTINE
IN DAM.PRO129 CALL PREVIOUSLY NAMED PHASE 1/2/9 'TO' ROUTINE
IN DAM.NTABS/DAM DAM UNIT * TABLE GOES IN SAME SEO W/ FORTRAN I/O

UTILITIES FOR PHASE 0.1.2.9 COMMANDS

SE0 S-READS*. (S-PRO0129)
IN DAM.READS 'READ' INTO UNIT 5 BUFFER
IN DAM.GETS 'GET' FREE-FORMAT FIELD FROM UNIT 5 BUFFER
IN DAM.WRNS PROCESS WARNING DIAGNOSTIC FOR UNIT 5 FIELD
IN DAM.SPANS ENABLE/DISABLE SPANNING FOR UNIT 5

SE0 S-OPNCLPR*. (S-PRO0129)
IN DAM.OPRPRD OPEN ALT PRT FILES
IN DAM.CLOSPR CLOSE ALT PRT FILES

SE0 S-CALSWIN*. (S-PRO0129)
IN DAM.CALSYM CALIBRATE SYMBOL TABLE
IN DAM.CALSPA CALIBRATE PRINT/PLiT COEFFICIENTS FOR SPACING
IN DAM.CALWIN CALIBRATE WINDOW

PHASE 0.1.2.9 COMMANDS (FORTRAN I/O ALLOWED)

SE0 S-XOTEXI*. (S-READS.S-OPNCLPR.S-CALSWIN)
IN DAM.PROQQT PRTDET INITIALIZATION ROUTINE
IN DAM.PDPEXI PRTDET TERMINATION ROUTINE

SE0 S-PSTART*. (S-XOTEXI)
IN  OAM.PSTART  .  GENERAL  INITIALIZATION  ROUTINE

SEQ  $-OPN12N$  .  (S-XQTEX1)
IN  OAM.OPN12N  .  OPEN  INPUT  DETECTION  FILE(S)  (UNITS  21 ...  24)

SEQ  $-PSTOP$  .  (S-XQTEX1)
IN  OAM.PSTOP  .  GENERAL  TERMINATION  ROUTINE

SEQ  $-HELP$  .  (S-READS.S-OPNCLPR.S-CALSLWIN)
IN  OAM.KM00CL  .  CLEAR  WARNINGS/ERRORS
IN  OAM.KM00XP  .  EXPLAIN  PROGRAM/COMMAND
IN  OAM.KM0DF1  .  END  IF...FI  BLOCK
IN  OAM.KM0DF  .  BEGIN  IF...FI  BLOCK
IN  OAM.KM0NEX  .  PRINT  NEWS
IN  OAM.KM0NEX  .  CONDITIONALLY  PERFORM  NEXT  COMMAND
IN  OAM.KM0OFF  .  TURN  OFF  MODE  SWITCH(ES)
IN  OAM.KM0ON  .  TURN  ON  MODE  SWITCH(ES)
IN  OAM.KM0PEE  .  PEEK
IN  OAM.KM0POK  .  POKE
IN  OAM.KM0TIM  .  PRINT  CLOCK  TIME  &  CHARGE  TIME

SEQ  $-GEOMETRY$  .  (S-READS.S-OPNCLPR.S-CALSLWIN)
IN  OAM.KM0HER  .  GET/CHECK  TRANSVERSE  MERCATOR  CENT  MERIDIAN
IN  OAM.KM0SPA  .  GET/CHECK  WINDOW  SPACING
IN  OAM.KM0TIC  .  GET/CHECK  TICK  INTERVALS
IN  OAM.KM0WIN  .  GET/CHECK  WINDOW  ENVELOPE/VERTICES
IN  OAM.KM0ZON  .  GET/CHECK  UTM  PROJECTION  ZONE

SEQ  $-SPECS$  .  (S-READS.S-OPNCLPR.S-CALSLWIN)
IN  OAM.KM0TCH  .  GET/CHECK  DETECTION  CHANNELS
IN  OAM.KM0DEN  .  GET/CHECK  DENSITY  LIMITS
IN  OAM.KM0HEA  .  GET/CHECK  PAGE  HEADING(S)
IN  OAM.KM0RI  .  GET/CHECK  WINDOW  ORIGIN
IN  OAM.KM0RAD  .  GET/CHECK  RADIANCE  LIMITS
IN  OAM.PR0019  .  DISPLAY  (PHASE  9)
IN  OAM.PR019  .  LIST  (PHASE  9)
IN  OAM.PR0P19  .  PICTURE  (PHASE  9)

SEQ  $-HISC$  .  (S-READS.S-OPNCLPR.S-CALSLWIN)
IN  OAM.KM0COL  .  GET/CHECK  COLORS
IN  OAM.KM0COP  .  GET/CHECK  NUMBER  OF  OUTPUT  COPIES
IN  OAM.KM0INT  .  GET/CHECK  INTENSITY
IN  OAM.KM0PAQ  .  SKIP  TO  TOP  OF  NEXT  PAGE
IN  OAM.KM0PRI  .  GET/CHECK  PRINTER  SPECIFICATIONS
IN  OAM.KM0REN  .  RENUMBER  (GET/CHECK  NEW  WINDOW  SEQUENCE  NUMBER)
IN  OAM.KM0SYM  .  GET/CHECK  SYMBOLS

SEQ  $-EXEC$  .  (S-READS.S-OPNCLPR.S-CALSLWIN)
IN  OAM.KM0XXX  .  MACRO  COMMANDS
IN  OAM.KM0ADD  .  DYNAMIC  $ADD$
IN  OAM.KM0AS  .  DYNAMIC  $AS$
IN  OAM.KM0BRKPT  .  DYNAMIC  $BRKPT$
IN  OAM.KM0F  .  DYNAMIC  $FREE$
IN  OAM.KM0LO  .  DYNAMIC  $LOO$

SEQ  $-DISLISPIC$  .  (S-READS.S-OPNCLPR.S-CALSLWIN)
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAN.PRODIS . DISPLAY (PHASE 0)
IN DAN.PRORIS . LIST (PHASE 0)
IN DAN.PROPIC . PICTURE (PHASE 0)

SEQ S-HTSTA8. (S-READ5, 2-OPNCLPR. S-CALSWIN)
IN DAN.KMOCRO . CROSSTABULATE COLOR & INTENSITY
IN DAN.KMOTAB . TABULATE PREVIOUSLY DIS/LIS/PIC DATA

MONITOR FOR PHASE 3.4.5 COMMANDS -------------------------------

SEQ S-PRD345. S-PR00129
IN DAN.PROD345 . CALL PREVIOUSLY NAMED PHASE 3/4/5 'TO' ROUTINE

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

SEQ S-PROD13. (S-PRD345) . DISPLAY DETECTION DATA
IN DAN.PROD13

SEQ S-PROL13. (S-PRD345) . LIST DETECTION DATA
IN DAN.PROL13

SEQ S-PRP13. (S-PRD345) . PICTURE DETECTION DATA
IN DAN.PROP13

L-260
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM.PRTDET/VIRTUAL

L-281
HISTORY

E M SCHLOSSER  LEC  03/12/74  ORIGINAL CODE
E M SCHLOSSER  LEC  07/09/70  DELETE RETN K'S & NUMERIC OPTIONS
E M SCHLOSSER  LEC  01/30/79  ADD MACRO COMMANDS
J C CRISP      LEC  12/21/79  LIST, MERIDIAN, PEEK, POKE, RADIANCE,
                  REMSCO  05/16/80  SPACING, TIME, IF, FI COMMANDS
                  CHAN, COLOR, CROSS, INTENSITY, PICT, TAB

METHOD

RETRIEVE NEXT USER COMMAND. VALIDATE IT AND CALL ITS SUBROUTINE.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

READS     3 READ PUNCHED CARD OR TERMINAL INPUT
GETAL    8 GET ALPHABETIC COMMAND
INTREC    8 INTEGER-ICE    8 INTEGER-CHAR-EQUIV FOR CHARACTER
WARNS    8 PRINT/LOG WARNING MESSAGE
PRO...   8 DEDICATED SUBROUTINE FOR COMMAND ... (SEE BELOW)
KNO...   8 COMMON SUBROUTINE FOR COMMAND ... (SEE BELOW)

EXCEPTIONS

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

GLOBAL DECLARATIONS

INCLUDE NULCST.LIST     8 DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS
INTEGER KOMO    0 FIRST 3 CHAR OF USER COMMAND (BLANK AFTER DONE)
INTEGER LSTAT  0 READS STATUS ('EOF' MEANS END-OF-FILE)
INTEGER KASE   0 MODIFIED 1-C-E OF FIRST CHAR OF COMMAND

PROCEDURE

CALL PREVIOUSLY NAMED SUBROUTINE

CALL TRACSE
CALL NAMSUD 0 CALL TO NULSUB DOES NOTHING

READ COMMAND FROM UNIT 3 (CARD READER OR TERMINAL)

KOMO. 'NUL' 0 IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
IF(INBATCH.NE.0) CALL READSILSTAT,' ' 0 FILL BUFFER, BLANK CUE MSG
IF(INBATCH.EQ.0) CALL READSLSTAT, NULSTAT 0 FILL BUFFER, NO CUE MSG
IF(LSTAT.NE.'EOF') IF(KOMO.NE.'EOF') CALL GETSAL(KOMO,3). NULSTAT 0 GET 3 ALPHA CHAR

CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
KASE=ICE(KOMO)-ICE('A')+1 0 A TO Z = 1 TO 26

CASE STATEMENT ON MODIFIED 1-C-E OF COMMAND'S FIRST CHARACTER

IF((KASE.LT.1).OR.(KASE.GT.26)) KASE=27 0 NOT ALPHA
GO TO 1
0 481,482,483,484,485,486,487,488,489,490,491,
1 441,442,443,444,445,446,447,448,449,450,
2 481,482,483,484,485,486,487,
& .KASE

DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOMO TO BLANK

401 CONTINUE 8*** A
402 CONTINUE 8*** B
GO TO 600

403 CONTINUE 8*** C
IF(KOMO.EQ.'CHA') CALL DETCHA(KOMO) 0 CHANNEL (DETECTION)
IF(KOMO.EQ.'CLE') CALL KHCLE(KOMO) 0 CLEAR
IF(KOMO.EQ.'COL') CALL KHCOL(KOMO) 0 COLOR
IF(KOMO.EQ.'CROS') CALL KHCROS(KOMO) 0 CROSSTAB
GO TO 600

404 CONTINUE 8*** D
IF(KOMO.EQ.'DEN') CALL KHDEN(KOMO) 0 DENSITY

L-263
IF(KOND.EQ.'DIS') CALL PRODIS(KOND)  G0 TO 800
C
400 CONTINUE 8000 G
IF(KOND.EQ.'EOF') CALL PROEXI(KOND)  8 END-OF-FILE CAUSES EXIT
IF(KOND.EQ.'EXIT') CALL PROEXI(KOND)  8 EXIT
IF(KOND.EQ.'EXP') CALL KMDEXP(KOND)  8 EXPLAIN
GO TO 800
C
408 CONTINUE 8000 F
IF(KOND.EQ.'FI') CALL KNDFI(KOND)  8 FI (ENDIF)
GO TO 800
C
407 CONTINUE 8000 E
GO TO 800
C
408 CONTINUE 8000 H
IF(KOND.EQ.'HEA') CALL KDHEA(KOND)  8 HEADING
GO TO 800
C
409 CONTINUE 8000 I
IF(KOND.EQ.'IF') CALL KNDIF(KOND)  8 IF
IF(KOND.EQ.'INT') CALL KNDINT(KOND)  8 INTENSITY
GO TO 800
C
410 CONTINUE 8000 J
411 CONTINUE 8000 K
GO TO 800
C
412 CONTINUE 8000 L
IF(KOND.EQ.'LIST') CALL PROLIST(KOND)  8 LIST
GO TO 800
C
413 CONTINUE 8000 M
IF(KOND.EQ.'MER') CALL KMDMER(KOND)  8 MERIDIAN
GO TO 800
C
414 CONTINUE 8000 N
IF(KOND.EQ.'NEW') CALL KMDNEW(KOND)  8 NEWS
IF(KOND.EQ.'NEXT') CALL KMDNEXT(KOND)  8 NEXT
GO TO 800
C
415 CONTINUE 8000 O
IF(KOND.EQ.'OFF') CALL KMDOFF(KOND)  8 OFF
IF(KOND.EQ.'ON') CALL KMDON(KOND)  8 ON
IF(KOND.EQ.'ORI') CALL KMDORI(KOND)  8 ORIGIN
GO TO 800
C
416 CONTINUE 8000 P
IF(KOND.EQ.'PAQ') CALL KMDPAQ(KOND)  8 PAGE
IF(KOND.EQ.'PCE') CALL KMDPCE(KOND)  8 PEEK
IF(KOND.EQ.'PIC') CALL PRODIC(KOND)  8 PICTURE
IF(KOND.EQ.'POK') CALL KMDPOK(KOND)  8 POKE
IF(KOND.EQ.'PRI') CALL KMDPRI(KOND)  8 PRINTER
GO TO 800
C
DAN PACKAGE APPENDIX L
MAIN PROGRAM Routines

417 CONTINUE B**** O
GO TO 800

418 CONTINUE B**** R
IF(KOND.EQ.'RAD') CALL KMDRAD(KOND) 8 RADIANCE
IF(KOND.EQ.'REN') CALL KMDREN(KOND) 8 RENUMBER
GO TO 800

419 CONTINUE B**** S
IF(KOND.EQ.'SPA') CALL KMDSPA(KOND) 8 SPACING
IF(KOND.EQ.'SYM') CALL KMDSYM(KOND) 8 SYMBOLS
GO TO 800

420 CONTINUE B**** T
IF(KOND.EQ.'TAB') CALL KMDTAB(KOND) 8 TABULATE
IF(KOND.EQ.'TIM') CALL KMDTIM(KOND) 8 TIME
GO TO 800

421 CONTINUE B**** U
422 CONTINUE B**** V
GO TO 800

423 CONTINUE B**** W
IF(KOND.EQ.'WIN') CALL KMDWIN(KOND) 8 WINDOW
GO TO 800

424 CONTINUE B**** X
425 CONTINUE B**** Y
GO TO 800

426 CONTINUE B**** Z
IF(KOND.EQ.'ZON') CALL KMDZON(KOND) 8 ZONE
GO TO 800

427 CONTINUE B**** NOT ALPHABETIC
IF(KOND.EQ.'BAD') CALL KMDBAD(KOND) 8 SADD
IF(KOND.EQ.'SAD') CALL KMDSAD(KOND) 8 SADD
IF(KOND.EQ.'UBS') CALL KMDUBS(KOND) 8 SADD
IF(KOND.EQ.'SRK') CALL KMDSRK(KOND) 8 SRKPT
IF(KOND.EQ.'SFR') CALL KMDSFR(KOND) 8 SFR
IF(KOND.EQ.'SLO') CALL KMDSLO(KOND) 8 SLO

428 CONTINUE B**** NOT ALPHABETIC
IF(KOND.EQ.'BAD') CALL KMDBAD(KOND) 8 SADD
IF(KOND.EQ.'SAD') CALL KMDSAD(KOND) 8 SADD
IF(KOND.EQ.'UBS') CALL KMDUBS(KOND) 8 SADD
IF(KOND.EQ.'SRK') CALL KMDSRK(KOND) 8 SRKPT
IF(KOND.EQ.'SFR') CALL KMDSFR(KOND) 8 SFR
IF(KOND.EQ.'SLO') CALL KMDSLO(KOND) 8 SLO

C IF COMMAND WAS NOT FOUND, TRY MACRO-COMMAND
C
800 IF(KOND.NE. 'PRO') '1 KOND=PRO-3 AND 3 CHAR OF PROG NAME PLUS --'
IF(KOND.NE. 'PR') CALL KMDXXX(KOND) 8 MACRO COMMAND HANDLER
C
C COMMAND IS INVALID IF STILL NOT FOUND
C
IF(KOND.NE. 'PRO') '1 CALL WARNI 'INVALID COMMAND --'
C
C FORCE ALL FORTRAN I/O ROUTINES INTO SAME SEO AS PROG00 (NEVER PERFORMED)
C
IF(KOND.EQ.'JUN.') READ(899.899) KOND

L-289
C
C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C
RETURN
END
SUBROUTINE PRD129: Call Phase 1/8/8 Subroutines for PRODET

NAME SUBRUTINE TO CALL

-----------------------------

CALL NAMSUB

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

RETURN
END
SUBROUTINE PRO3451  & CALL PHASE 3/4/5 SUBROUTINES FOR PRTOET
   I NAMSUB)  & NAME OF SUBROUTINE TO CALL

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

L-269
SUBROUTINE PRO015 ( & DISPLAY DETECTION FILE(S) (PHASE 0) )
U KOMO) & I: FIRST 3 CHAR OF COMMAND & 0: SPACES

HISTORY

E H SCHLOSSER LEC 10/05/78  ORIGINAL CODE
E H SCHLOSSER LEC 06/20/78  DELETE RETN K & MAKE EXTERNAL SUB
E H SCHLOSSER LEC 03/12/79  RADIANCE & CLASS DETECTION FILES
J C CRISP  LEC 12/21/79  SPLIT OUT PRO013 & PRO019

METHOD

CALIBRATE SYMBOLS/SPACING/WINDOW, OPEN PRINT FILES, DISPLAY HEADINGS.
THEN NAME PRO013 TO GENERATE BODY OF DISPLAY.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

EXTERNAL PRO000.  NULSUB
EXTERNAL PROD29.  PRO019
EXTERNAL PRO349.  PRO013

EXCEPTIONS

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
GLOBAL DECLARATIONS
-------------------

INCLUSION OF COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUSION OF COMMON/DEFINE FOR ALTERNATE PRINT FILES
INCLUSION OF COMMON ERTS SCENE PARAMETERS
INCLUSION OF COMMON CLASSIFICATION SPECTRAL LIMITS
INCLUSION OF COMMON Adjustment/Registration PARAMETERS
INCLUSION OF COMMON TICK/FREQ/FACTOR TABLE
INCLUSION OF COMMON SYMBOL TABLE
INCLUSION OF DEFINE STRUCTURE OF WINDOW PACKETS
INCLUSION OF COMMON OUTPUT WINDOW PACKETS
INCLUSION OF DEFINE NULL CHARACTER STRING
INCLUSION OF DEFINE PRDDEF PARAMETERS

LOCAL DECLARATIONS
------------------

INTEGER KNTEMP
INTEGER NTEMP
INTEGER NPRLIN, NPRCOL
INTEGER NPRCIF
INTEGER LUNCH

PROCEDURE
----------

CALL TRACE

GET DISPLAY TYPE

CALL GETSKNI(KNTEMP, (31), NULCST)  # ALLOW (BUT IGNORE) OPTIONAL TYPE
CALL NVIATOI (PRD345, PRD013)  # NEXT CALL IS TO PRD013

DRAIN SPECS FOR CURRENT COMMAND
300 CALL GETSTAT(INTERN, -1.1, 'EXTRA DISPLAY SPECIFICATION --')

CC
CHECK RADIANCE LIMITS
CC
IF(LCVLOI.GT.LCVHII) CALL NWARN('NO RADIANCE LIMITS')
IF(MDATAEM.EQ.0) GO TO 900 DATA/CHECKOUT MODE
CC
CALIBRATE SYMBOLS/SPACING/WINDOW
CC
CALL CALSYM
CALL CALSPA
CALL CALWIN(0)
CC
OPEN PRINT FILE(S) IF NOT OPEN, CLEAR WINDOW NUMBER & RESET PAGE NUMBER
CC
IF(NNOTOTL.NE.0) GO TO 900
IF(WINHOLD.LT.0) CALL OPRAV0 & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
WINHOLD=ABS(WINHOLD)
NPAGE=0
CC
COMPUTE SIZE OF PRINT WINDOW
CC
NPRLIN=FIX(PDOWHN(LLIN,WMX))=FIX(PDOWHN(LLIN,WMIN))+1
NPRLIN=FIX(PDOWHN(NCOL,WMX))=FIX(PDOWHN(NCOL,WMIN))+1
NPRLIN=FIX(KPAGEMAX) & NUMBER OF PRINT COLUMNS PER FILE EXCLUDING
LEFT MARGIN AND 2 HEAT LINE CHARs
CC
CHECK WIDTH OF PRINT WINDOW
CC
NTMAX=1+(NPRLIN)/NPRLIN
IF(NMAX.GT.MAXW) CALL NWARN('WINDOW TOO WIDE')
CC
CHECK FOR DIAGNOSTICS
CC
IF(NNOTOTL.NE.0) GO TO 900
IF(MBATCH.EQ.0).AND.(NCISYM.EQ.0)) CALL NDNOTE('SYMBOLS NOT OVERPRINTED ON DEMAND TERMINAL')
CC
PRINT WINDOW HEADING FOR UNIT 6
CC
WRITE(6,15) WINHOLD,TERMA
15 FORMAT('WINDOW NUMBER 'J3.'DISPLAY'6X.4A6)
CALL IORRA(6)
CALL I0CPRO(6)
CC
PRINT WINDOW HEADING FOR ALTERNATE PRINT FILE(S)
CC
LUNALT=10

L-271
DO 460 N=1,NPRCOL,NPRCIF
   CALL MDUNIT( LUNALT)
   WRITE(LUNALT,415) NNDOW,MTERAL
   CALL IDERT( LUNALT)
   CALL IDCPRD( LUNALT)
   LUNALT=LUNALT+1
460 CONTINUE

C ANY DIAGNOSTICS???
C
900 IF(NDOTTL.EQ.0) GO TO 990
   IF(NDATAC.NE.0) CALL NVIATO( PRD000,NULSUB)  & DATA/CHECKOUT
   IF(NDATAC.EQ.0) CALL NVIATO( PRD129,PRDD19)
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KONT=9
   RETURN
END
SUBROUTINE PROO13 & DISPLAY DETECTION FILE(S) (PHASE 3)

HISTORY

E H SCHLOSSER LEC 10/05/75 ORIGINAL CODE
E H SCHLOSSER LEC 06/20/78 DELETE RETN K4 MAKE EXTERNAL SUB
E H SCHLOSSER LEC 03/12/79 RADIANCE & CLASS DETECTION FILES
J C CRISP LEC 12/21/79 MAKE SEPARATE ROUTINE (FROM PROOIS)
J C CRISP LEMSCO 05/18/80 USE CHANNEL(S) SPECIFIED BY LIMCH
J C CRISP LEMSCO 08/18/80 ADD 4 WORDS TO PRINT BUFFER

METHOD

INITIALIZE LOW AND HIGH PRINT LINES AND COLUMNS. GENERATE AND
PRINT TOP SAMPLE SCALE AND BORDER. INITIALIZE LOW AND HIGH
ADJUSTED LINE. CALL READ2N TO READ LINE. MASK NON-TRIVIAL
WINDOW. RESAMPLE/SCREEN/COUNT FREQUENCY/SYMBOLIZE LINE.
OUTPUT LINE. GENERATE AND PRINT BOTTOM SAMPLE SCALE AND
BORDER. NAME PROD19 AS 'TO' ROUTINE FOR WRAP-UP OF DISPLAY
PROCESSING.

MACHINE-DEPENDENT CODE

UTILIZES UNIVAC EXEC 8 ER PRNTAS
INTERNAL ROUTINE SANSCL ASSUMES 8 CHARS TO AN INTEGER BIN

EXTERNAL REFERENCES

ANP    A ADJUSTED COORD FOR PRINT/PLOT COORD
READ2N A READ SCAN LINE FROM DETECTION FILE(S)
MSPKX  A MASK NON-TRIVIAL WINDOW
PROVFI A PRINT/OVERPRINT Files
MOFI  A PRINT/COUNT 'FATAL ERROR' MESSAGES
NVIATO A NAME 'VIA' 'TO' SUBROUTINES
ERPRTA A WRITE TO ALTERNATE PRINT FILES
CS4V1N A CHARACTER STRING FOR INTEGER
DOUBLE PRECISION C8V4CS A VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH
DOUBLE PRECISION CB4V1N A VARIABLE-LENGTH CHAR STRING FOR INTEGER

EXTERNAL PROO129, PROD19
EXTERNAL GETASY, GETBYT, GETICE, GETINT, GETNUL & 'TO' ROUTINES GET BIN

EXCEPTIONS

STATUS

L-273
### MAIN PROGRAMS/Routines

#### FROM READEH NEAT LINE CHAR PRINT CELL SYMBOLS DIAGNOSTIC ACTION

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>'EOF'</strong></td>
<td>'1'</td>
<td>NO DATA ('1'+'1')</td>
<td>NONE</td>
<td>PRINT LINE</td>
<td></td>
</tr>
<tr>
<td><strong>'BADR'</strong></td>
<td>'?'</td>
<td>NO DATA ('1'+'1')</td>
<td>NONE</td>
<td>PRINT LINE</td>
<td></td>
</tr>
<tr>
<td><strong>'BAOF'</strong></td>
<td>N/A</td>
<td>N/A</td>
<td>FATAl</td>
<td>RETURN</td>
<td></td>
</tr>
<tr>
<td><strong>'OFL'</strong></td>
<td>N/A</td>
<td>N/N</td>
<td>FATAl</td>
<td>RETURN</td>
<td></td>
</tr>
</tbody>
</table>

### Global Declarations

- INCLUDE KOMXOT.LIST
- INCLUDE KOMKLS.LIST
- INCLUDE KOMSYM.LIST
- INCLUDE WINOEr.LIST
- INCLUDE KOMOWW.LIST
- INCLUDE KOMTBL.LIST
- INCLUDE PRODEF.LIST
- INCLUDE PXBDEF.LIST
- INCLUDE MAXINT.LIST

### Local Declarations

- PARAMETER NDBUF5 = 2  // NUMBER OF DETECTION BUFFERS
- INTEOEERS IN DET BUF = #INTS PREAMBLE + (#BINS+3)/4 + #EXTRA INTS
- PARAMETER NWI0BF = 2*(PXBINS-1) + (3548+2+3)/4 + 27
- WDS PRT BUF=WDS PREAM+(MAX FILES*(MAX COLS/PG-NEAT LN COLS-MARO COLS))/4
- PARAMETER NWIPBF=(PXBINS-1)+HALT H + (KPA0H1-2-41) + 4
- PARAMETER NFROSZ=128 // WDS PER CHANNEL IN LOCAL FREO TABLE
- PARAMETER NFROCH=5 // CHANNELS IN LOCAL FREO TABLE
- INTEOEER NDTBUF=(NW10BF,NDBUF5) // ARRAY OF DETECTION BUFFERS
- INTEOEER NDBF // DETECTION BUFFER NUMBER
- INTEOEER NDBFI // HIGHEST DETECTION BUFFER NUMBER TO USE
- INTEOEER NRTBUF=NWIPBF // PRINT BUFFER
- INTEOEER NFROI (NFROSZ,NFROCH) // LOCAL FREO TABLE (SCOPE INCLUDES
- INTEOEER ISPUE (NFRSOZ) // INTERNAL ROUTINE RESYM
- INTEOEER IPMUN, I PMAX // MINIMUM AND MAXIMUM PRINT COLUMN
- INTEOEER I PLMIN, I PLMAX // MINIMUM AND MAXIMUM PRINT LINE
- REAL ADJLMM, ADJSM // ADJUSTED LINE AND SAMPLE
- INTEOEER ML100L,ML100H,ML100S // MSA LINE+100: LOW, HIGH, SPACING
- INTEOEER MSAL, MSLIN // MSA LINE NUMBER
- INTEOEER MSALS, MSASH // LOW AND HIGH MSA SAMPLE
- INTEOEER I STAT // I/O STATUS
- INTEOEER NTLCHR // NEAT LINE CHARACTER
- INTEOEER NROLCS,MRORCS // LEFT & RIGHT MARGIN CHAR STRING
- INTEOEER NPLIN,NPRCOL // NUMBER OF PRINT LINES AND COLUMNS
- INTEOEER LASTLN // LAST LINE READ

### Procedure

L-274
CALL TRACE

DETERMINE HOW MANY BUFFERS TO USE

IF(NLINCN.GT.NOBUFS) CALL MONOTE('TOO MANY CHANNELS REQUESTED -- FIRST',NOBFH1,2).

DETERMINE HOW MANY BUFFERS TO USE

INITIALIZE MINIMUM AND MAXIMUM PRINT LINES AND COLUMNS

INITIALIZE NUMBER OF PRINT LINES AND COLUMNS

CLEAR LOCAL FREQUENCY TABLE

INITIALIZE LOW AND HIGH LINES AND SPACING

GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE BEFORE 1ST PRINT LINE

READ. MASK. RESAMPLE. SCREEN. SYMBOLIZE AND PRINT SCAN LINES

iplin=iplmin
LASTLN=MAXINT
DO 350 ML100=ML1000,ML1000
  NSALIN=NL100/100
  CALL ANP (ADJLN,ADJSAM, FLOAT(IPLIN,FLOAT(IPCHIN))
  NSASLO=ADJSAM
  CALL ANP (ADJLN,ADJSAM, FLOAT(IPLIN,FLOAT(IPCHNAX))
  NSASH=ADJSAM
  IF (NSALIN.EQ.LASTLN) GO TO 320 0 ALREADY IN BUFFER(S)
  NTL=CHM=4
  DO 310 NSBF=1,NSBFH
    CALL READN (MOTBUF(1),NSBF), (NMIDBP), ISTAT.
    NSALIN.LINCH(NSIDF), NSASLO, NSASH)
    IF (ISTAT.EQ.'BAD') NTL=CHM=7.
    IF (ISTAT.NE.'BAD') AND.
      (ISTAT.NE.'OFL') GO TO 310
    CALL MDFAT(I, CPSICS(ISTAT,1), 4)
      'WHILE READING DETECTION FILE'
    CALL ERROR (109, 1, E, G,
      '*I/O ERROR - IGNORE OUTPUT**'
    GO TO 900
  310 CONTINUE
  LASTLN=NSALIN
  CALL MKPIX (MOTBUF(1), MDFAT(1,1))
  IF (MDFAT(PXINT,1), EQ. 'Bt') NOR.
    (MDFAT(PX1NT,1), EQ. 'CHR') OR.
    (MDFAT(PXINT,1), EQ. 'INT') OR.
    (MDFAT(PXINT,1), EQ. 'NUL') GO TO 330
    CALL MDFAT(I, 'INVALID BIN TYPE '
      CPSICS(MDFAT(PXINT,1), 1), 3), ' IN PROD13')
  GO TO 900
  330 IF (MDFAT(PXINT,1), EQ. 'Bt') CALL RESYM (IPTBF, IPLIN).
    (IPCHIN), (IPMAX), MDFAT, (NMIDBF), NSBF, GETBYT)
    IF (MDFAT(PXINT,1), EQ. 'CHR') CALL RESYM (IPRTBF, IPLIN).
      (IPCHIN), (IPCHNAX), MDFAT, (NMIDBF), NSBF, GETCHR)
    IF (MDFAT(PXINT,1), EQ. 'INT') CALL RESYM (IPRTBF, IPLIN).
      (IPCHIN), (IPCHNAX), MDFAT, (NMIDBF), NSBF, GETINT)
    IF (MDFAT(PXINT,1), EQ. 'NUL') CALL RESYM (IPRTBF, IPLIN).
      (IPCHIN), (IPCHNAX), MDFAT, (NMIDBF), NSBF, GETNUL)
    CALL CSTN (NRORCS, (1), (6), NSALIN,4, 0)
    MRORCS=NRORCS
    IF (MBATCH.EQ.0) AND. (NPRLN.LE.64) AND. (NPRLC,LE.64)
      CALL PROVF (6, MROCSS, 4, 0, NTLCHR, '090000', 7, IPRBF)
      CALL PROVF (10, MROCSS, 4, MROCSS, 4, NTLCHR, 7, '000000', 7, IPRBF)
      IPLIN=IPLIN+1
  350 CONTINUE

C C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE AFTER LAST PRINT LINE
C CALL SANSCL (IPRTBF, (IPMAX,1), (IPCHIN), (IPCHNAX))
  IF (MBATCH.EQ.0) AND. (NPRLN.LE.64) AND. (NPRLC,LE.64)
    CALL PROVF (6, '****', '****', '****', '****', '****', '****', '****', 7, IPRBF)
    CALL PROVF (10, '****', '****', '****', '****', '****', '****', '****', '****', 7, IPRBF)
    CALL PROVF (10, '****', '****', '****', '****', '****', '****', '****', '****', 7, IPRBF)
C MOVE DATA FROM LOCAL FREQ TABLE TO COMMON TABLE (REPLACING TICKS)

C
C KTBLY='FREQ'
KTBLYM='NDON
C DO 600 I=1,NLINCH
DO 950 K=1,NFRGZ
KFREQ(K,1)=NFREQ(K,1)
650 CONTINUE
600 CONTINUE
DO 700 I=1,10
DO 850 K=1,15
KFRCRO(I,K)=MAXINT
850 CONTINUE
700 CONTINUE

C NEXT CALL IS TO PRD19

C 900 CALL NVIATO ( PRD129,PRD19)
C RETURN

C INTERNAL SUBROUTINE SAMSCL ( GENERATE SAMPLE SCALE AND BORDER
C
C 1 PRBF 2 PRINT BUFFER
1 IPRINT 2 PRINT LINE
1 IPCMIN 2 MINIMUM PRINT COLUMN
1 IPCMAX 2 MAXIMUM PRINT COLUMN

C METHOD
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING. SET BUFFER
PREAMBLE. ENCODE SAMPLE NUMBERS AND PUT COLON STRING, AND
COLON IN PRINT BUFFER.

C MACHINE-DEPENDENT CODE
C
C ASSUMES 6 CHARMS PER INTEGER BIN

C EXTERNAL REFERENCES
C
C AHP 2 ADJUSTED COORD FOR PRINT/PLT COORD
C PUTCNR 2 PUT CHAR IN CHAR STRING
C CST4IN 2 CHARACTER STRING FOR INTEGER

C GLOBAL DECLARATIONS
C
L-277
INCLUDE KONMMM.LIST 0 OUTPUT WINDON PACKETS
INCLUDE WINDEF.LIST 0 DEFINE WINDON PACKETS
INCLUDE PXKDBF.LIST 0 DEFINE BUFFER STRUCTURE

LOCAL DECLARATIONS

INTEGER IPRTBF(11) 0 ARGUMENT
REAL ADJSAM 0 ADJUSTED SAMPLE
INTEGER IPBIN 0 POINTER TO PRINT BIN
INTEGER MSASAM 0 SAMPLE NUMBER
INTEGER MS100L,MS100M,MS100S 0 MSA SAMPLE=100: LOW,HIGH,SPACING

PROCEDURE

INITIALIZE LOW AND HIGH SAMPLES AND SPACING
CALL AVP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCHIN))
MS100L=ADJSAM+100.
CALL AVP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCHAX))
MS100H=ADJSAM+100.
MS100S=MSADIM(MSAD, MS100)

SET PREAMBLE POINTERS AND BIN POINTER
IPRTBF(PXRECH)=0
IPRTBF(PXLIN)=IPLIN
IPRTBF(PXCHAN)=0
IPRTBF(PXQUAL)=0
IPRTBF(PXBINT)='INT'
IPRTBF(PXLBIN)=1
IPRTBF(PXLCOL)=IPCHIN
IPRTBF(PXHBIN)=IPCHAX-IPCHIN+1
IPRTBF(PXHCOL)=IPCHAX
IPRTBF(PXQIN)=0
IPRTBF(PXQODA)=0
IPRTBF(PXJ01)=0
IPRTBF(PXHJO1)=0
IPBIN=IPRTBF(PXLBIN)-1

ENCODE SAMPLE NUMBER AND PUT COLON STRING AND COLON IN BUFFER
DO 100 MS100=MS100L,MS100M,MS100S
MSASAM=MS100/100
CALL PUTCHR(IPRTBF(PXBINS+IPBIN)(11),':')
CALL CS14N(IPRTBF(PXBINS+IPBIN)(2),0,MSASAM,0,:')
CALL PUTCHR(IPRTBF(PXBINS+IPBIN)(8),':')
IPBIN=IPBIN+1
100 CONTINUE

RETURN
INTERNAL SUBROUTINE RESYN ( 8 RESAMPLE/SCREEN/COUNT FREQUENCY/SYMBOLIZE
5 IPRTBF, 8 PRINT BUFFER
5 IPRINT, 8 PRINT LINE
5 IPCHIN, 8 MINIMUM PRINT COLUMN
5 IPCHAX, 8 MAXIMUM PRINT COLUMN
5 IHTBUF, 8 ARRAY OF DETECTION BUFFERS (ALL SAME SINTYPE)
5 NWCBF, 8 NUMBER OF WORDS IN ONE BUFFER
5 NOBUF'S, 8 NUMBER OF BUFFERS
5 GETBIN, 8 ROUTINE TO GET BIN VALUE--GETBYT,GETICE,GETINT,GETNUL

METHOD
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. COUNT FREQUENCY, AND SYMBOLIZE.

EXTERNAL REFERENCES
A4P 8 ADJUSTED MSS COORD FOR PRINT/PLT COORD

GLOBAL DECLARATIONS

INCLUDE KOMM.WLIST 8 COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMM.WLIST 8 COMMON FREQUENCY/TICK TABLE
INCLUDE KOMK.SL1ST 8 COMMON CLASSIFICATION INFO
INCLUDE PKBDF.LIST 8 DEFINE BUFFER STRUCTURE
INCLUDE PKSWM.LIST 8 COMMON SYMBOL TABLE
INCLUDE PKDEF.LIST 8 DEFINE WINDOW PACKETS

LOCAL DECLARATIONS

PARAMETER NUMBS=8 8 NUMBER OF MSS BUFFERS
INTEGER HTBUF(MFREE.MFREE.OUTBUFS) 8 ARGUMENT
INTEGER IPRTBF(11) 8 ARGUMENT
INTEGER MS10D(NUMBS) 8 BIN NUMBER OF SAMPLE 0 FOR EACH MSS BUFFER
INTEGER MS100L,MS100H,MS100S 8 MSS SAMPLE=100: LOW, HIGH, SPACING
INTEGER MS100S 8 MSS SAMPLE NUMBER
INTEGER IPXN 8 PRINT BUFFER BIN POINTER
REAL ADJSA 8 ADJUSTED SAMPLE NUMBER
INTEGER IPX1,IPX2 8 PIXEL VALUE FOR EACH BUFFER

L-279
PROCEDURE

INITIALIZE LOW AND HIGH SAMPLES AND SPACING

CALL A4P (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCMIN))
MS100L = ADJSAM + 100.
CALL A4P (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCMAX))
MS100H = ADJSAM + 100.
MS100S = MSAMMIN(MSAM, MSH100)

SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER

DO 150 NUMBUF = 1, NUMBUFS
NBINSO(NUMBUF) = HDTBUF(PXLSAM, NUMBUF) - HDTBUF(PXLSAM, NUMBUF)
150 CONTINUE

SET PREAMBLE POINTERS AND BIN POINTER

IPRTBF(PXRECNI*MDTBUF(PXRECNI), IPLIN)
IPRTBF(PXLIN0) = IPLIN
IPRTBF(PXCH0) = 0
IPRTBF(PXH0) = IPMINT
IPRTBF(PXLF0) = 2
IPRTBF(PXCOL) = PCMINT
IPRTBF(PXHCOL) = PCMINT + 2
IPRTBF(PXH0) = PCMINT
IPRTBF(PXNOA0) = 0
IPRTBF(PXNOA0) = 0
IPRTBF(PXNOA0) = 0
IPRTBF(PXNOA0) = 0
IPRTBF(PXNOA0) = 0
IPRTBF(PXNOA0) = 0

RESAMPLE/SCREEN RADIANCE/COUNT FREQUENCY/LOOK UP SYMBOLS

DO 400 MS100 = MS100L, MS100H, MS100S
MSASAM = MS100/100

BUFFER 1

IF ((MSASAM.LT. HDTBUF(PXLSAM, 1)) .OR.
   MSASAM.GT. HDTBUF(PXHSAH, 1)) GO TO 350  & SAMPLE NOT IN BUFFER
   CALL GETBIN (IPIXI),
     HDTBUF(PXBINS, 1), MSASAM = NBINSO(1))
   IF (IPIXI .LE. HDTBUF(PXNOA, 1)) GO TO 350  & NO DATA
   IF (IPIXI .LT. LCVL0) .OR.
     (IPIXI .GT. LCVH)) GO TO 380  & OUT OF RAD LIMITS
   IF (NL1MC1.EQ.0) GO TO 320

BUFFER 2

IF ((MSASAM.LT. HDTBUF(PXLSAM, 2)) .OR.

L-280
(MSASAM.GT.NOTBUF(PXMSAM.GT.2)) GO TO 350  
   & SAMPLE NOT IN BUFFER
   CALL GETBIN (1IPXL2,
   NOTBUF(PXBIN.GT.1),(MSASAM+NBINS0(2)))
   IF (1IPXLI.2.LT.LCVLO(2)).OR.
   (1IPXLI.2.GT.LCVHI(2)) GO TO 360  & OUT OF RAD LIMITS

C COUNT FREQUENCY AND SYMBOLIZE

   NREQ(IPXLI.2+1.2)=NREQ(IPXLI.2+1.2)+1
   NREQ(IPXLI.1+1.1)=NREQ(IPXLI.1+1.1)+1
   IPRTBF(PXBIN.1+IPBIN)=KSYM(IPXLI.1)
   GO TO 390
   350 IPRTBF(PXBIN.1+IPBIN)='.'  & NO DATA SYMBOL
   GO TO 390
   360 IPRTBF(PXBIN.1+IPBIN)=' '  & OUT OF RADIANCE LIMITS
   390 IPBIN=IPBIN+1
   400 CONTINUE

C

900 RETURN

END
SUBROUTINE PROO19 & DISPLAY DETECTION FILE(S) (PHASE 8)

HISTORY

- E M SCHLOSSER LEC 10/05/75 ORIgINAL CODE
- E M SCHLOSSER LEC 08/27/78 UPGRADE DOCUMENTATION
- E M SCHLOSSER LEC 05/01/79 DON'T INCREMENT WINDOW IF DIAGNOSTIC
- J C CRISP LEC 12/26/79 REVISE CODE IN PICTAB FOR PRDDET

METHOD

- CHECK DIAGNOSTIC COUNTERS AND PREPARE FOR NEXT DISPLAY.

MACHINE-DEPENDENT CODE

- NONE.

EXTERNAL REFERENCES

- MCA R TE
- MCCLRw

EXCEPTIONS

- NONE.

GLOBAL DECLARATIONS

- INCLUDE KOMXOT.LIST COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
- INCLUDE NULCSR.LIST DEFINE NULL CHARACTER STRING
- EXTERNAL PROO000, NULSUB

PROCEDURE

- CALL TRAC.

ON RETURN, CALL PROO000 TO GET COMMANDS

- CALL NVIATOI( PROO000, NULSUB)

L-282
C ANY DIAGNOSTICS???
C
900 IF(NDFATL.EQ.0) GO TO 920
   CALL MDNOTE( 'FATAL ERRORS -- NO DISPLAY GENERATED')
   GO TO 990
920 IF(NDWARN.EQ.0) GO TO 980
   CALL MDNOTE( 'PREVIOUS WARNINGS -- NO DISPLAY GENERATED')
   IF(M9AtCN.EQ.0) WRITE(8,925)
   925 FORMAT(*,'**TRY AGAIN!')
   CALL MDCLAIM( NULCST)
   GO TO 990
C
C PREPARE FOR NEXT DISPLAY
C
980 NWNDOM=NWNDOM+1
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
990 RETURN
END
SUBROUTINE PROEXII: A TERMINATION ROUTINE FOR PROGET
U KONDI & L: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY
---
E H SCHLOSSER LEC 08/12/75 DESIGN/COORD/TEST
J C CRISP LEC 12/31/78 PRINT DISPLAYS ONSITE

METHOD
-------
CONFIRM PROGRAM TERMINATION. ASK ABOUT PRINTER DISPLAYS. PRINT
DISPLAYS ONSITE, IF REQUESTED. ELSE DELETE ALTERNATE PRINT FILES.
TERMINATE PROGRAM.

MACHINE-DEPENDENT CODE
-------------------
NONE

EXTERNAL REFERENCES
-------------------
DELETEPR: DELETE ALTERNATE PRINT FILES
READS: "FILL BUFFER FOR UNIT S"
CLOSPR: "CLOSE (AND PRINT) ALTERNATE PRINT FILES"
PSTOP: "PROGRAM TERMINATION"

EXCEPTIONS
---------
NONE

GLOBAL DECLARATIONS
-------------------
INCLUDE KOMXOT.LIST "COMMON PROGRAM EXECUTION SWITCHES, COUNTERS"
INCLUDE KOMNER.LIST "COMMON ERTS SCENE PARAMETERS"
INCLUDE NULLCST.LIST "DEFINE NULL CHARACTER STRING"

LOCAL DECLARATIONS
-------------------
INTEGER NORY: "'N' OR 'Y' RESPONSE TO PRINT DISPLAYS ONSITE"

PROCEDURE
C --------

C CALL TRACE

C CONFIRM PROGRAM TERMINATION

WRITE(8,125)
125 FORMAT(4X,'**PROGRAM TERMINATION**')

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 MDWARN('FATAL ERROR(S) ENCOUNTERED -- DISPLAY(S) ARE DEFECTIVE')
   WRITE(6,185)
185 FORMAT('OUTPUT DISPLAY(S) ON LINE PRINTER?')
   IF(KOMO.NE.'EOF') CALL READS(LSTAT,')
   NORY='N'
   CALL GETSKM(NORY,(1), NULCST)
   IF(NORY.NE.'Y') GO TO 700

C

C PRINT DISPLAY(S) ONSITE

C 200 NHNDOW=MAX(0,NHNDOW-1)
   WRITE(6,245) NHNDOW
245 FORMAT(IX,14, 'DISPLAY(S) PRINTED')
   CALL CLOSPr
   GO TO 900

C

C DON'T PRINT DISPLAY(S) ONSITE

C 700 CALL DLETPr

C

C TERMINATE PROGRAM

C 900 CALL PSTOP(NULCST)

C

C PSTOP DOES NOT RETURN

END
SUBROUTINE PROL1S: A LIST DETECTION FILE(S) (PHASE 0)
U KOND) & 1: FIRST 3 CHARs OF COMMAND 0: SPACES

HISTORY
-------
J C CRISP	 LEC	 09/21/79	 REQUIREMENTS
J C CRISP	 LEC	 10/13/79	 ALGORITHM DESIGN
J C CRISP	 LEC	 12/26/79	 ALGORITHM CODING

METHOD
-------
CHECK/CALIBRATE SPECS. GENERATE LIST HEADINGS.
THEN NAME PROL13 TO GENERATE BODY OF LIST.

MACHINE-DEPENDENT CODE
----------------------
UTILIZES UNIVAC FORTRAN V FUNCTION BOOL.

EXTERNAL REFERENCES
-------------------
GETSKH	 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN	 GET INTEGER DATA FIELD FROM UNIT 5
HOMARN	 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
NVIATO	 NAME 'VIA' 'TO' SUBROUTINES
CALSPA	 CALIBRATE TRANSFORMATION COEFFICIENTS FOR SPACING
CALWIN	 CALIBRATE WINDOW ENVELOPES
OPRPRO	 OPEN ALTERNATE PRINT FILE(S)
IDENT	 IDENTIFY ERTS SCENE
IDCPRO	 IDENTIFY CURRENT COMMAND SPECS FOR PRDDET
HOUNIT	 WRITE HEADING LINE(S) AT TOP OF NEXT PAGE
WARNS	 SUBMIT WARNING FOR MISSING/INVALID FIELD FROM UNIT 5

EXTERNAL PRD00.	 NULSUB
EXTERNAL PRD129.	 PROL19
EXTERNAL PRD345.	 PROL13

EXCEPTIONS
-----------
1. 'LIST' MAY NOT BE A DEFAULT COMMAND.
2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE LIST.
3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS:

<table>
<thead>
<tr>
<th>CONDITION</th>
<th>ACTION</th>
<th>DIAGNOSTIC</th>
</tr>
</thead>
</table>
| L-288
C PROCESSING DEFAULT COMMANDS
C (NWWOWN=0)
C LIMIT CHANNEL VALUE RANGE IS NULL
C DATA/CHECKOUT NODE
C WARNING(S) OR FATAL ERROR(S)
C
C GLOBAL DECLARATIONS
C-------------------
INCLUDE KOMXOT.LIST
INCLUDE KOMALT.LIST
INCLUDE KOMNER.LIST
INCLUDE KOMKLS.LIST
INCLUDE KOMTBL.LIST
INCLUDE WINDEF.LIST
INCLUDE KOMOWN.LIST
INCLUDE NULCST.LIST

C LOCAL DECLARATIONS
C
INTEGER KHTEMP
INTEGER INTEMP
INTEGER NPPRLIN
INTEGER NPRLIN
INTEGER LUNALT
INTEGER NPPCIF
INTEGER NODMAX
INTEGER NOD
INTEGER PPOTOW(2,WW0)

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

C GET LIST TYPE
C
KTBLTY=' NUL' & MARK OLD FREQ TABLE AS DESTROYED
IF(NWWOWN.EQ.0) CALL MDWARN("INVALID DEFAULT COMMAND")
CALL GETSKM(KHTEMP,31,NULCST)
CALL NVIATO(PRO34S,PROL13) & NEXT CALL IS TO PROL13

C DRAIN SPECS FOR CURRENT COMMAND
C
300 CALL GETSIN(INTEMP, *1,-1,'EXTRA LIST SPECIFICATION --')
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C CHECK RADIANCE LIMITS
C
IF (LCVAL1 .GT. LCVH1) CALL MDWARN( "NO RADIANCE LIMITS")
IF (MOATAC .NE. 0) GO TO 900  & DATA/CHECKOUT MODE

C CALIBRATE SPACING
C
CALL CALSPA

C SAVE THE PHYSICAL PPD VERTEX COLUMNS
C
NOMAX = BOOL (PPDOM (WUSED, WHEAD))
DO 350 NOD = WVER, NOMAX
PPDTOM (WCOL, NOD) = PDWM (WCOL, NOD)
350 CONTINUE

C TRANSFORM TO LOGICAL PPD VERTEX COLUMNS FROM PHYSICAL
C
DO 375 NOD = WVER, NOMAX
PPDOM (WCOL, NOD) = PDWM (WCOL, NOD) / 3
375 CONTINUE

C CALIBRATE OUTPUT WINDOW ENVELOPE USING LOGICAL PPD VERTEX COLUMNS
C
CALL CALWIN( 0.1 )

C RESTORE THE PHYSICAL PPD VERTEX COLUMNS
C
DO 400 NOD = WVER, NOMAX
PPDOM (WCOL, NOD) = PDWM (WCOL, NOD)
400 CONTINUE

C CHECK FOR DIAGNOSTICS
C
IF (NOTOTL .NE. 0) GO TO 900

C OPEN PRINT FILE(S) IF NOT OPEN. CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
IF (NWNDOW .LT. 0) CALL OPRPRO  & OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
NWNDOW = IABS (NWNDOW)
NPAGE = 0

C COMPUTE LOGICAL SIZE OF PRINT WINDOW FROM LOGICAL PPD ENVELOPE
C
NPRLIN = IFIX (PPDOM (MLIN, WMAX)) - IFIX (PPDOM (MLIN, WMIN)) + 1
NPRCOL = IFIX (PPDOM (WCOL, WMAX)) - IFIX (PPDOM (WCOL, WMIN)) + 1
NPRCIF = (KPAGE - 2) / 3  & NUMBER OF LOGICAL PPD COLUMNS PER FILE

L-288
C
C CHECK WIDTH OF PRINT WINDOW
C
NITMAX=1+(NPRCOL)/(NPRCIF)
IF(NITMAX.GT.MALT) CALL HDWARN('WINDOW TOO WIDE')
C
C CHECK FOR DIAGNOSTICS
C
IF Nd TotL NE 0) GO TO 900
C
C PRINT WINDOW HEADING FOR UNIT 6
C
WRITE(6,415) Nwindow, MTERAL
415 FORMAT(' WINDOW NUMBER ', 'J3', '6X', 'LIST', '6X', '4A6')
CALL IDERT( 6)
CALL IOCPRD( 6)
C
C PRINT WINDOW HEADING FOR EACH ALTERNATE PRINT FILE IN LOGICAL PPD ENVELOPE
C
LUNALT=10
DO 480 N=1,NPRCOL,NPRCIF
   CALL HDUNIT( N, LUNALT)
   WRITE(LUNALT,415) Nwindow, MTERAL
   CALL IDERT( LUNALT)
   CALL IOCPRD( LUNALT)
   LUNALT=LUNALT+1
480 CONTINUE
C
C ANY DIAGNOSTICS???
C
900 IF Nd TotL EQ 0) GO TO 990
   IF(MDATAAC NE 0) CALL NVIATo( PRD000, NULSUB) & DATA/CHECKOUT
   IF(MDATAAC EQ 0) CALL NVIATo( PRD129, PROL19)
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KOMD=''
   RETURN
END

L-289
SUBROUTINE PROLIS & LIST DETECTION FILE(S) (PHASE 3)

HISTORY

J C CRISP	 LEC	 09/21/79	 REQUIREMENTS
J C CRISP	 LEC	 10/13/79	 ALGORITHM DESIGN
J C CRISP	 LEC	 12/20/79	 ALGORITHM CODING
J C CRISP	 LEMSCO	 09/16/80	 USE CHANNEL(S) SPECIFIED BY LINCH
J C CRISP	 LEMSCO	 09/16/80	 ADD 4 WORDS TO PRINT BUFFER

METHOD

- INITIALIZE LOW AND HIGH PRINT LINES AND COLUMNS. GENERATE AND PRINT TOP SAMPLE SCALE AND BORDER. INITIALIZE LOW AND HIGH ADJUSTED LINE. CALL READEN TO READ LINE. MASK NON-TRIVIAL WINDOW. RESAMPLE/SCREEN/COUNT FREQUENCY.
- GENERATE AND PRINT BOTTOM SAMPLE SCALE AND BORDER.
- NAME PROLIS AS 'TO' ROUTINE FOR WRAP-UP OF LIST PROCESSING.

MACHINE-DEPENDENT CODE

ASSUMES 8 CHARs TO AN INTEGER BIN.

EXTERNAL REFERENCES

- ADJUSTED COORD FOR PRINT/PLT COORD
- READEN 8 READ SCAN LINE FROM DETECTION FILE(S)
- GETCHR 8 GET CHARACTER FROM CHARACTER STRING
- MSKPIX 8 MASK NON-TRIVIAL WINDOW
- PROFIN 8 PRINT/OVERPRINT FILES
- MODLAL 8 PRINT/LOO/COUNT 'FATAL ERROR' MESSAGES
- NVIAD 8 NAME 'VIA' 'TO' $ROUTINES
- ERPTA 8 WRITE TO ALTERNATE PRINT FILES
- CSTWIN 8 CHARACTER STRING FOR INTEGER
- DOUBLE PRECISION COSCS 8 VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH
- DOUBLE PRECISION COSIN 8 VARIABLE-LENGTH CHAR STRING FOR INTEGER
- EXTERNAL PROLIS. PROLIS
- EXTERNAL RESCRN, GETBYT, GETC, GETINT, GETNUL 8 ROUTINE TO GET BIN VALUE

EXCEPTIONS

- STATUS FROM
- READEN HEAT LINE CHAR PRINT CELL SYMBOLS DIAGNOSTIC ACTION

L-290
**Global Declarations**

- INCLUDE KOMXOT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
- INCLUDE KOMKL.LIST 8 COMMON CLASSIFICATION INFO
- INCLUDE WINDEF.LIST 8 DEFINE STRUCTURE OF WINDOW PACKETS
- INCLUDE KOMHNL.LIST 8 COMMON OUTPUT WINDOW PACKETS
- INCLUDE KOMTL.LIST 8 COMMON TICK/FREQ TABLE
- INCLUDE PXBDEF.LIST 8 DEFINE BUFFER STRUCTURE
- INCLUDE PROODEF.LIST 8 DEFINE PROD/DET PARAMETERS
- INCLUDE MAINTLIST 8 MAXIMUM INTEGER VALUE

**Local Declarations**

- PARAMETER NBUFFS=2 8 # OF DETECTION BUFFERS IN ARRAY
- INTEGERS IN DET BUF = #INTS PREAMBLE + (#BINS+3)/4 + #EXTRA INTS
- PARAMETER NH1DSF = 2*(#BINS-1) + 1540*2+31/4 + 27
- WDS PRT BUF=WDS PREAM+(MAX FILES*(MAX COLS/PX-MEATLN COLS-MARG COLS)+1)*4
- PARAMETER NH1PSF=(#BINS-1)*MALTNI + (#PXBINS-2 - 41)*4
- PARAMETER NFROS2=128 8 # WDS PER CHANNEL IN LOCAL FREQ TABLE
- PARAMETER NFROCH=5 8 # CHANNELS IN LOCAL FREQ TABLE
- INTEGER NH1DSF(NH1DSF,NH1DSF) 8 ARRAY OF DETECTION BUFFERS
- INTEGER NHDF = DETECTION BUFFER NUMBER
- INTEGER NH1DFH = HIGHEST DETECTION BUFFER NUMBER TO USE
- INTEGER IPRTSF(NH1DSF) 8 PRINT BUFFER
- INTEGER NFROCH(NFROS2,NFROCH) 8 LOCAL FREQUENCY TABLE (SCOPE INCLUDES
- INTEGER IPLIN 8 PRINT LINE
- INTEGER IPCIN,IPCMAX 8 MINIMUM, MAXIMUM PRINT COLUMN
- REAL ADJL,ADJSAM 8 ADJUSTED LINE AND SAMPLE
- INTEGER HL1000,HL100M,ML100 8 MSA LINE=100: LOW, HIGH, SPACING
- INTEGER MSA100 8 MSA LINE NUMBER
- INTEGER 1STAT 8 I/O STATUS
- INTEGER HL1CHR 8 NEAT LINE CHARACTER
- INTEGER MROCS,MROCS 8 LEFT & RIGHT MARGIN CHAR STRING
- INTEGER IPLIN,IPMX 8 MINIMUM AND MAXIMUM PRINT LINE
- INTEGER ML100 8 MSA LINE=100
- INTEGER MSAL0,MSAH 8 LOW & HIGH MSA SAMPLE NUMBER
- INTEGER NPRIN,NPRCOL 8 # PRINT LINES, COLUMNS
- INTEGER LASTLN 8 LAST LINE READ

**Procedure**

- L-201
CALL TRACE

DETERMINE HOW MANY BUFFERS TO USE

NDBFH1=MIN(NLINCH,NDBUFS)
NLIMCH=NLINCH+NLIMCH
IF(NLINCH.GT.NDBUFS) CALL HDNOTE
   'TOO MANY CHANNELS REQUESTED -- FIRST'.CBS41(NDBFHI,2).
   & 'CHANNELS PROCESSED'

INITIALIZE MINIMUM AND MAXIMUM PRINT LINES AND COLUMNS

PLMIN=PPDOWM(NLIN,WMIN)
PLMAX=PPDOWM(NLIN,WMAX)
PCMIN=PPDOWM(NCOL,WMIN)
PCMAX=PPDOWM(NCOL,WMAX)

CALCULATE THE NUMBER OF PRINT LINES, COLUMNS TO BE PRINTED

NPRLIN=PLMAX-PLMIN+1
NPRLON=(PCMAX-PCMIN+1)+3

CLEAR LOCAL FREQUENCY TABLE

DO 250 I=1,NLIMCH
   DO 260 K=1,NFREQS
      NFREQ(I,K)=0
   260 CONTINUE
   250 CONTINUE

INITIALIZE LOW AND HIGH LINES AND SPACING

CALL ANP (ADJLIN,ADJSAM, FLOAT(PLMIN),1.)
ML100=ADJLIN+100.
CALL ANP (ADJLIN,ADJSAM, FLOAT(PLMAX),1.)
ML100M=ADJLIN+100.
ML100S=ML100M/MLIN.WSP1001

GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE BEFORE 1ST PRINT LINE

300 CALL SAMSCL (IPRT0F,PLMIN-1,PLMIN+1,PCMIN,PCMAX,1)
   IF (ICOL.EQ.0).AND.(NPRLIN.LE.64).AND.(NPRLON.LE.64)
      CALL PROVF1 (0. '*.*.0.0.*1111*'.IPRT0F)
      CALL PROVF1 (10. '*.*.0.0.*1111*'.IPRT0F)

READ, MASK, RESAMPLE, SCREEN, COUNT FREQUENCY AND PRINT SCAN LINES

PLMIN=PLMIN
LASTLN=MAXINT

L-292
DO 300 NL100=NL100NL100NL100
MSLIN=NL100/100
CALL AQV (ADJIN, ADJJSN), FLOAT (PLIN), FLOAT (PCCH)
MSASLO=ADJJSN
CALL AQV (ADJIN, ADJJSN), FLOAT (PLIN), FLOAT (PCCH)
MSASLI=ADJJSN
IF (MSALIN.EQ.LASLIN) GO TO 320 8 ALREADY IN BUFFER
NLCHR='.'
DO 310 NOBF=1.NOBFHI
CALL READEN (HOTBTF (I, NOBF), (NHIDDBF), ISTAT.
MSALIN.LINCH (NOBF), MSASLO, MSALSHI)
IF (ISTAT.EQ. 'BADF') NLCHR='.'
IF (ISTAT.NE. 'BADF').AND.
(NSTAT.NE. 'OFL') GO TO 310
CALL HOFATL:COS4CS (ISTAT, 1,4).
" WHILE READING DETECTION FILE"
CALL ERPRTA ('10', '2,0',
'1/O ERROR = IGNORE OUTPUT')
GO TO 300

310 CONTINUE
320 LASTLN=MSALIN
CALL MSKPIX (HOTBTF (I, 1), HOTBTF (I, 1))
IF ((HOTBTF (PXBINT, I), EQ. 'BYT').OR.
(HOTBTF (PXBINT, I), EQ. 'CHR').OR.
(HOTBTF (PXBINT, I), EQ. 'NT').OR.
(HOTBTF (PXBINT, I), EQ. 'NUL') ) GO TO 330
CALL HOFATL: "INVALID BIN TYPE",
COS4CS (HOTBTF (PXBINT, 1), 1,3), ' IN PROLI3"
GO TO 300
330 IF (HOTBTF (PXBINT, 1), EQ. 'BYT') CALL RESCN (IPRTBF, (IPLIN).
(IPCHIN), (IPCHAK), HOTBTF (NHIDDBF), (NDBUS).GETBYT)
IF (HOTBTF (PXBINT, 1), EQ. 'CHR') CALL RESCN (IPRTBF, (IPLIN).
(IPCHIN), (IPCHAK), HOTBTF (NHIDDBF), (NDBUS).GETCHR)
IF (HOTBTF (PXBINT, 1), EQ. 'NT') CALL RESCN (IPRTBF, (IPLIN).
(IPCHIN), (IPCHAK), HOTBTF (NHIDDBF), (NDBUS).GETNT)
IF (HOTBTF (PXBINT, 1), EQ. 'NUL') CALL RESCN (IPRTBF, (IPLIN).
(IPCHIN), (IPCHAK), HOTBTF (NHIDDBF), (NDBUS).GETNUL
CALL CSTLIN (MRGCS, 11, 161, MSALIN, 4, '0')
MRGCS=MRGCS.
IF (HMBATCH.EQ.01) ANP (NPRIN.LE. 64) ANP (NPRCOL.LE. 64)
CALL PROFI (8, MRGCS, 4, '0', NLCHR, '1', '0', '1', IPRBF)
CALL PROFI (10, MRGCS, 4, MRGCS, 4, '0', NLCHR, '1**', IPRBF)
IPLIN=IPLIN+1
350 CONTINUE

C GENERATE AND PRINT SAMPLE SCALE AND BORDER FOR LINE AFTER LAST PRINT LINE
CALL SAMSCN (IPRTBF, (IPLIN)+1, (IPCHIN), (IPCHAK))
IF (HMBATCH.EQ.01) ANP (NPRIN.LE. 64) ANP (NPRCOL.LE. 64)
CALL PROFI (8, '4, '0, '1', '1', IPRBF)
CALL PROFI (10, '1, '4, '0, '1', '1, IPRBF)
CALL PROFI (10, '1**', '4, '1**', '0, '3**, IPRBF)

C MOVE DATA FROM LOCAL FREQ TABLE TO COMMON TABLE

L-293
C

KTBLTY='FREQ'
KTBLNM=NNDNN
DO 500 I=1,NLINCH
   DO 550 K=1,NFREQZ
      KFREQ(K.I)=NFREQ(K.I)
   550 CONTINUE
500 CONTINUE
DO 700 I=1.10
   DO 650 K=1.15
      KFREQO(I.K)=MAXINT
650 CONTINUE
700 CONTINUE

C NEXT CALL IS TO PROL19
C
900 CALL NVIATO ( PROL19, PROL19)
RETURN

C
C
C INTERNAL
C
C
C SUBRINTNE SAMSCL ( I GENERATE SAMPLE SCALE AND BORDER
C
C PARAMETERS
C
C ( IPRINTF, A PRINT BUFFER
C ( IPLIN, A PRINT LINE
C ( IPCMIN, A MINIMUM PRINT COLUMN (LOGICAL)
C ( IPCMAX, A MAXIMUM PRINT COLUMN (LOGICAL)
C
C HISTORY
C
C J C CRISP LEC 12/26/79 REQUIREMENTS
C J C CRISP LEC 12/26/79 ALGORITHM DESIGN
C J C CRISP LEC 12/26/79 ALGORITHM CODING

C
C
C
C METHOD
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING, SET BUFFER
C PREAMBLE, ENCODE SAMPLE NUMBERS AND PLT COLON STRING, AND
C COLON IN PRINT BUFFER.
C NOTE: 5 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN.
C
C MACHINE-DEPENDENT CODE
C
C ASSUMES 6 CHARS PER INTEGER BIN
C
C EXTERNAL REFERENCES

L-294
G

DAN PACKAOE APPENDIX L  PROL13

MAIN PROGRAMS/ROUTINES

"SIN PROGRAMS/ROUTINES OOE

C

C	 Atop	 A AOUSTEO COORO FOR PRINt/PLOT COOR
C	 PIITCHR	 i PUT CHAR IN CHAR STRING
C

C GLOBAL DECLARATIONS
C
C	 INCLUDE KOMOWN.LIST 8 COMMON OUTPUT WINDOW PACKETS
C	 INCLUDE WINDEF.LIST 8 DEFINE STRUCTURE OF WINDOW PACKETS
C	 INCLUDE PX80EF.LIST 8 DEFINE PIXEL BUFFER STRUCTURE
C
C LOCAL DECLARATIONS
C

INTEOER IPRTBF(1) 8 ARGUMENT
REAL AOJLIN,ADJSAM 8 ADJUSTED LINE, SAMPLE
INTEGER IPBIN 8 POINTER TO PRINT BIN
INTEGER MSASAM 8 SAMPLE NUMBER
INTEGER MS100L,MS100H,MS100S 8 MSA SAMPLE*100: LOW,HIGH,SPACING
C
C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
C
CALL A4P (AOJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCMIN))
MS100L=ADJSAM*100.
CALL A4P (AOJLIN,ADJSAM, FLOAT(IPLIN),FLOAT(IPCMAX))
MS100H=ADJSAM*100.
MS100S=MSAOWW(MSAM,WSP100)
C
C SET PREAMBLE POINTERS
C
IPRTBF(PXRECNI)=0
IPRTBF(PXLIN0)=IPLIN
IPRTBF(PXCHAN)=0
IPRTBF(PXQUAL)=0
IPRTBF(PXBIN1)=INT
IPRTBF(PXB1N)=2
IPRTBF(PXLBO1)=IPCMIN
C
C CONVERT LOGICAL PPD COLUMNS TO PHYSICAL PPD BINS
IPRTBF(PXHBIN1)=(1(IPCMAX-IPCMIN+1)+3)+1
C
IPRTBF(PXCOL1)=IPCMAX
IPRTBF(PXNO1)=0
IPRTBF(PXNOAD)=0
IPRTBF(PXLJ01)=0
IPTB(PXHJOI)-0
IPBIN=IPRTBF(PXLBIN)-1

BUILD SCALE AND BORDERS
(3 PHYSICAL PPO BINS PER LOGICAL PPD COLUMN)

DO 100 MS100=MS100L,MS100H,MS100S
MSASAM=MS100/100
IPRTBF(PXBIN+IPBIN+1)=0
IPBIN=IPBIN-1
CALL PUTCHR(IPTBFPXBINS+IPBIN+2),(1),*:"
CALL CSTYN (IPTBF(PXBIN+IPBIN+2),(2),4,MSASAM,'0') & '9999:'
CALL PUTCHR(IPTBF(PXBIN+IPBIN+2),(6),*:"

100 CONTINUE

RETURN TO CALLING ROUTINE

900 RETURN

INTERNAL
SUBROUTINE RESCRN I A RESAMPLE/SCREEN/COUNT FREQUENCY
O IPTB. A PRINT BUFFER
(IPLIN. A PRINT LINE
(IPCMIN. A MINIMUM PRINT COLUMN (LOGICAL)
(IPCMAX. A MAXIMUM PRINT COLUMN (LOGICAL)
(NMTBUF. A ARRAY OF DETECTION BUFFERS (ALL OF SAME BINTYPE)
(NWIOBUF. A NUMBER OF WORDS IN ONE BUFFER
(NOBUFFS. A NUMBER OF BUFFERS
(GETBIN) A ROUTINE TO GET BIN VALUE--GETBYT,GETICE,GETINT,GETNIL

HISTORY
J C CRISP LEC 12/28/79 REQUIREMENTS
J C CRISP LEC 12/28/79 ALGORITHM DESIGN
J C CRISP LEC 12/28/79 ALGORITHM CODING

METHOD
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.
MACHINE-DEPENDENT CODE

EXTERNAL REFERENCES

EXTERNAL

AJP

ADJUSTED MSS COORD FOR PRINT/ PLOT COORD

CSTWIN

CHARACTER STRING FOR INTEGER

GETICE

GET CHARACTER FROM CHARACTER STRING

EXCEPTIONS

NONE.

GLOBAL DECLARATIONS

PARAMETER NUM9FS=2

INTEGER N01DBF(N1DBF, NDBUFS) ARGUMENT

INTEGER PRTBF(1) ARGUMENT

INTEGER UPLOAD1, UPLOAD2 VALUE OF PIXEL FOR EACH MPX BUFFER

INTEGER RJBKDBUFS) BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER

INTEGER MS100L, MS100H, MS100S MSA SAMPLE*100: LOW, HIGH, SPACING

INTEGER MSASAM MSA SAMPLE NUMBER

INTEGER PPIXIN PRINT BUFFER BIN POINTER

REAL ADJLIN, ADJSAM ADJUSTED LINE AND SAMPLE

INTEGER MS100 MSA SAMPLE*100

INTEGER KSTPIX PIXEL VALUE CHARACTER STRING

PROCEDURE

CALL AJP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCHIN), MS100L, MSASAM*100).

CALL AJP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCHMAX), MS100H, MSASAM*100).

MS100S=MSADWH(WSAM, WSP100)
C SET NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
C
DO 150 NUMBUF=1,NDBUF
   NBINSO(NUMBUF)=NDBUF(PXLBIN.NUMBUF)-NDBUF(PXLSAM.NUMBUF)
150 CONTINUE
C
C SET PREAMBLE POINTERS AND BIN POINTER
C
   IPRTBF(PXRECN)=NDBUF(PXRECN.1)
   IPRTBF(PXLINE)=PLIN
   IPRTBF(PXCHAN)=0
   IPRTBF(PXQUAL)=0
   IPRTBF(PXBIN)="INT"
   IPRTBF(PXLBIN)=2
   IPRTBF(PXCOL)=PCMIN
C
C CONVERT LOGICAL PPD COLUMNS TO PHYSICAL PPD BINS
C
   IPRTBF(IPXBIN)=(IPCMAX-IPCMIN+1)*3+1
C
   IPRTBF(IPXCOL)=IPCMAX
   IPRTBF(IPXNOI)=0
   IPRTBF(IPXNOO)=0
   IPRTBF(IPXJOIN)=0
   IPRTBF(IPXJOIN)-IPRTBF(PXLBIN)-1
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.MDBUF(PXLSAM.1)).OR.
      & (MSASAM.OT.MDBUF(PXHSAM.1)) GO TO 350  A SAMPLE NOT IN BUFFER
   CALL GETBIN(IPXL1.
      & MDBUF(PXBSAM.1).MSASAM+NBINSO(1))
   IF (IPXL1.0E.MDBUF(PXNOA.1)) GO TO 350  A NO DATA
   IF (IPXL1.0E.LCVLO1).OR.
      & (IPXL1.0E.LCVHLO1)) GO TO 360  A OUT OF RAD LIMITS
   IF (NLINCH-1.EQ.0) GO TO 320
C
C BUFFER 2
C
   IF (MSASAM.LT.MDBUF(PXLSAM.2)).OR.
      & (MSASAM.OT.MDBUF(PXHSAM.2))) GO TO 350  A SAMPLE NOT IN BUFFER
   CALL GETBIN(IPXIL2.
      & MDBUF(PXBSAM.2).MSASAM+NBINSO(2))
   IF (IPXIL2.0E.LCVLO2).OR.
      & (IPXIL2.0E.LCVHLO2)) GO TO 360  A OUT OF RAD LIMITS
C
C COUNT FREQUENCY
C
   NFREQ(IPXIL2.1,2)=NFREQ(IPXIL2.1,2)+1
C

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

320   NFREQ(IPIXL1+1,1)=NFREQ(IPIXL1+1,1)+1

C   INSERT RADIANCE VALUE INTO PRINT BUFFER
C   (3 PHYSICAL PPD BINS PER LOGICAL PPD COLUMN)
C
   CALL CST4IN(KSTPIX,(1:3,1)
   CALL GETCHR(IPRTBF(PXBINS+IPBIN=0), KSTPIX,(1))
   CALL GETCHR(IPRTBF(PXBINS+IPBIN=1), KSTPIX,(2))
   CALL GETCHR(IPRTBF(PXBINS+IPBIN=2), KSTPIX,(3))
   IF(IPRTBF(PXBINS+IPBIN=0).EQ.'0') IPRTBF(PXBINS+IPBIN+0)='':
   GO TO 390

C   INSERT 'NO DATA' SYMBOLS INTO PRINT BUFFER
C
   IPRTBF(PXBINS+IPBIN=0)='':
   IPRTBF(PXBINS+IPBIN+1)='':
   IPRTBF(PXBINS+IPBIN+2)='':
   GO TO 390

C   INSERT 'NO INFO' SYMBOLS INTO PRINT BUFFER
C
   IPRTBF(PXBINS+IPBIN=0)='':
   IPRTBF(PXBINS+IPBIN+1)='':
   IPRTBF(PXBINS+IPBIN+2)='':
   IPBIN=IPBIN+3
   390 CONTINUE

C   RETURN TO CALLING ROUTINE
C
   900 RETURN

END
SUBROUTINE PROL19 - DISPLAY DETECTION FILE(S) (PHASE 9)

HISTORY
-------
E H SCHLOSSER LEC 02/23/79 ORIGINAL CODE
E H SCHLOSSER LEC 09/24/78 UPGRADE DOCUMENTATION
E H SCHLOSSER LEC 05/01/79 DON'T INCREMENT WINDOW IF DIAGNOSTIC
J C CRISP LEC 12/28/78 REVISE CODE IN PICTAB FOR PTDDET

METHOD
------
CHECK DIAGNOSTIC COUNTERS AND PREPARE FOR NEXT LIST.

MACHINE-DEPENDENT CODE
-----------------------
NONE.

EXTERNAL REFERENCES
---------------------
MONOTE
MOCCLR

EXCEPTIONS
---------
NONE.

GLOBAL DECLARATIONS
---------------------
INCLUDE KOMXOT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULLST.LIST & DEFINE NULL CHARACTER STRING
EXTERNAL PROD000, NULSUB

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

ON RETURN, CALL PROD000 TOGET COMMANDS
CALL NVIATO PROD000,NULSUB
C ANY DIAGNOSTICS???
C
900 IF(NDFATL.EQ.0) GO TO 920
   CALL MDNLOG('FATAL ERRORS -- NO LIST GENERATED')
   GO TO 990
920 IF(NDFARN.EQ.0) GO TO 980
   CALL MDNLOG('PREVIOUS WARNINGS -- NO LIST GENERATED')
   IF(IMBATCH.EQ.0) WRITE(6,925)
925 FORMAT(4X,'***TRY AGAIN!')
   CALL MDCLR(NULCST)
   GO TO 990
C
C PREPARE FOR NEXT WINDOW
C
980 NWNDOW=NWNDOW+1
C
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
990 RETURN
END
SUBROUTINE PROPIC: 8 PICTURE DETECTION FILE(S) (PHASE 0)
U(100ND) 8 I: FIRST 3 CHARS OF COMMAND 0; SPACES

HISTORY
E M SCHLOSSER  LEC  05/17/79  REQUIREMENTS & DESIGN
E M SCHLOSSER  LEC  11/06/79  STUBBED
E M SCHLOSSER  LEMSCO  06/08/80  IMPLEMENTED

METHOD
CHECK/CALIBRATE SPECS. CLEAR SCREEN. GENERATE PICTURE HEADINGS.
THEN NAME PROP13 TO GENERATE BODY OF PICTURE.

MACHINE-DEPENDENT CODE
NONE.

EXTERNAL REFERENCES
GETSKH  GET CHARACTER STRING DATA FIELD FROM UNIT S
GETSI  GET INTEGER DATA FIELD FROM UNIT S
MDMARN  PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
MDNOTE  PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
MVIAO  NAME 'VIA' 'TO' SUBROUTINES
CALCMA  CALIBRATE CHANNELS
CALCLO  CALIBRATE COLOR/INTENSITY SYMBOLS
CALSPA  CALIBRATE TRANSFORMATION COEFFICIENTS FOR SPACING
CALMIN  CALIBRATE WINDOW ENVELOPES
CROPW  CROP OUTPUT WINDOW
EAPRN  WRITE ASCII IMAGE TO PRIMARY OUTPUT DEVICE
OAPPRD  OPEN ALTERNATE PRINT FILE(S)
IDERT  IDENTIFY ERTS SCENE
IDCPRD  IDENTIFY CURRENT COMMAND SPECS FOR PRDET
MOUNT  WRITE HEADING LINE(S) AT TOP OF NEXT PAGE
WARNS  SUBMIT WARNING FOR MISSING/INVALID FIELD FROM UNIT S

VIA TO EXTERNAL PRD000, NULSUB
EXTERNAL PRD129, PROP19
EXTERNAL PRD345, PROP13

EXCEPTIONS

1. 'PICTURE' MAY NOT BE A DEFAULT COMMAND.
2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE PICTURE.

3. THE FOLLOWING EXCEPTION CONDITIONS PRODUCE THE FOLLOWING RESULTS

<table>
<thead>
<tr>
<th>CONDITION</th>
<th>ACTION</th>
<th>DIAGNOSTIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESSING DEFAULT COMMANDS</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>KLSTYP IN COMMON UNDEFINED</td>
<td>USE COMMON KLSTYP</td>
<td>NONE</td>
</tr>
<tr>
<td>KLSTYP SPECIFICATION MISSING</td>
<td>USE COMMON KLSTYP</td>
<td>WARNING</td>
</tr>
<tr>
<td>KLSTYP SPECIFICATION INVALID</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>EXTRA SPECIFICATION</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>LIMIT CHANNEL VALUE RANGE IS NULL</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>COLOR MODE SWITCH NOT ON</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>BATCH RUN</td>
<td>NONE</td>
<td>WARNING</td>
</tr>
<tr>
<td>DATA/CHECKOUT MODE</td>
<td>&quot;TO&quot; ROUTINE IS NULSUB</td>
<td>NONE</td>
</tr>
<tr>
<td>WARNING(S) OR FATAL ERROR(S)</td>
<td>&quot;TO&quot; ROUTINE IS PROP19</td>
<td>NONE</td>
</tr>
</tbody>
</table>

GLOBAL DECLARATIONS

LOCAL DECLARATIONS

PROCEDURE

CHECK IF COMMAND IS LEGAL

IF(WINDOW.EQ.0) CALL MOWARN("INVALID DEFAULT COMMAND")
IF(BATCH.NE.0) CALL MOWARN("PICTURE COMMAND NOT ALLOWED IN BATCH")
IF(NCGR0.EQ.0) CALL MOWARN("PICTURE COMMAND NOT ALLOWED (COLOR MODE NOT ON")

L-303
GET PICTURE TYPE

KTBLTY='NUL' 8 MARK OLD FREQ TABLE AS DESTROYED
CALL GETSH(KHTEMP, NULCST) 8 ALLOW (BUT IGNORE) OPTIONAL PICT TYPE
CALL NVIATO( PRO346, PROPI3) 8 NEXT CALL IS TO PROPI3

DRAIN SPECS FOR CURRENT COMMAND

300 CALL GETSIN(INTEMP, "*I-1", 'EXTRA PICTURE SPECIFICATION --')

CHECK RADIANCE LIMITS

IF(LCVLO1.GT.LCVHI) CALL MDWARMN ("NO RADIANCE LIMIT!")
IF(MDWTAC.NE.0) GO TO 900 8 DATA/CHECKOUT MODE

CALIBRATE COLORS/SPACING/WINDOW

CALL CALCOL
CALL CALSPA
CALL CALWIN1 0.1

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

IF(NOTOTL.NE.0) GO TO 900
IF(NWNOOW.LT.0) CALL OPRPRO 8 OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
NWNOOW=ABS(NWNOOW)
NPAOE=0

CROP OUTPUT WINDOW TO FIT INPUT WINDOW & COLOR SCREEN

CALL CROPWIN(KSLINE.KSCOLM)
IF(NOTOTL.NE.0) GO TO 900

COMPUTE SIZE OF PRINT/PLT DEVICE (COLOR SCREEN) WINDOW

NPRLIN=IFIX(PPDOWW(WLIN.WMAX)) - IFIX(PPDOWW(WLIN.WMIN)) + 1
NPRCOL=IFIX(PPDOWW(WCOL.WMAX)) - IFIX(PPDOWW(WCOL.WMIN)) + 1

CHECK FOR DIAGNOSTICS

IF(NOTOTL.NE.0) GO TO 900

CLEAR SCREENS AND PRINT WINDOW HEADING

CALL EAPRNT(0,1.KSON) 8 ROUTE UNIT 8 OUTPUT TO COLOR SCREEN
CALL EAPRNT(0,1.KSCLER) 8 CLEAR COLOR SCREEN
WRITE(6,415) HINDD.Mternal
415 FORMAT(' ',WINDOW NUMBER', '*J2.8X,'PICTURE', *6X,4A8)
CALL EAPPNT(0.1,KSOFF)  & ROUTE UNIT 6 OUTPUT TO B&W SCREEN
WRITE(6,415) HINDD.NTERNAL
CALL IDENT( 6)  
CALL IDCPRO( 6)

C
C ANY DIAGNOSTICS???
C
900 IF(NDOTOL.EQ.0) GO TO 990
   [IF(MDATAC.NE.0) CALL NVIATO( PRD000.NULSUB) 8 DATA/CHECKOUT
   [IF(MDATAC.EQ.0) CALL NVIATO( PRD129,PRP19)
C
C RETURN FOR CALL TO NAMED SUBROUTINE
C
990 KOND=-1  
RETURN
END
SUBROUTINE PROPIS & PICTURE DETECTION FILE(S) (PHASE 3)

HISTORY

E N SCHLOSSER LEC 06/17/70 REQUIREMENTS & DESIGN
E N SCHLOSSER LEC 11/06/70 STUBBED
E N SCHLOSSER LEMSCO 06/08/80 IMPLEMENTED

METHOD

INITIALIZE LOW AND HIGH PPD LINES AND COLUMNS. INITIALIZE LOW AND HIGH ADJUSTED LINE. CALL READEN TO READ LINE. MASK NON-TRIVIAL WINDOW. REAMPLE/SCREEN/COUNT FREQUENCY/SYMBOLIZE LINE. OUTPUT LINE. NAME PROPIS AS 'TO' ROUTINE FOR WRAP-UP OF PICTURE PROCESSING.

EXTERNAL REFERENCES

AMP 8 ADJUSTED COORD FOR PRINT/PLOT COORD
READEN 8 READ SCAN LINE FROM DETECTION FILE(S)
MSKPIX 8 MASK NON-TRIVIAL WINDOW
MDFATL 8 PRINT/LOG/COUNT 'FATAL ERROR' MESSAGES
NVIATO 8 NAME 'VIA' 'TO' SUBROUTINES
EAPRINT 8 WRITE ASCII IMAGE TO PRIMARY OUTPUT DEVICE
INTEGER NINVB 8 NUMBER OF INTEGERS FOR NUMBER OF BYTES
DOUBLE PRECISION COSIS 8 VARIABLE-LENGTH CHAR STRING FOR FIXED-LENGTH
DOUBLE PRECISION COSIN 8 VARIABLE-LENGTH CHAR STRING FOR INTEGER

EXTERNAL PR0129. PROPIS
EXTERNAL RESCOL. GETBYTES, GETICE, GETINT, GETNUL 8 ROUTINE TO GET BIN VALUE

EXCEPTIONS

STATUS FROM 8 PPD CELL SYMBOLS 8 DIAGNOSTIC 8 ACTION
READEN "EOF" NO DATA (GRAY) 8 NONE 8 WRITE LINE
"BADR" NO DATA (GRAY) 8 NONE 8 WRITE LINE
"BADF" N/A 8 FATAL 8 RETURN
"OFL" N/A 8 FATAL 8 RETURN

L-308
GLOBAL DECLARATIONS

INCLUDE KOMDOT.LIST  0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST  0 COMMON CLASSIFICATION INFO
INCLUDE KOMSYM.LIST  0 COMMON SYMBOL TABLE
INCLUDE KOMKS.LIST  0 COMMON COLOR SCREEN PARAMETERS
INCLUDE WINDFL.LIST  0 DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMOWN.LIST  0 COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMTBL.LIST  0 COMMON TICK/FREQ TABLE
INCLUDE PRODEF.LIST  0 DEFINE PRODEF PARAMETERS
INCLUDE PXDEF.LIST  0 DEFINE PIXEL BUFFER STRUCTURE
INCLUDE MAXINT.LIST  0 MAXIMUM INTEGER VALUE

LOCAL DECLARATIONS

PARAMETER NDBUF=2  0 # OF DETECTION BUFFERS IN ARRAY
INTEGERS IN DET BUF = #_INTS PREAMBLE = (#_INS+3)/4  0 # EXTRA INTS
PARAMETER NHIDOF = 2*(PXBSINS-1) = (3540-3)/4  0 27
#HDS COLOR/INTEIENCY BUF = HDS PREAM + 256
PARAMETER NMKOF=(PXBSINS-1)  0 256
PARAMETER NFRGCH=6  0 WDS PER CHANNEL IN LOCAL FREQ TABLE
PARAMETER NFRGSZ=128  0 CHANNELS IN LOCAL FREQ TABLE
INTEGER MDTBUF(NWIDOF,NDBUF)  0 ARRAY OF DETECTION BUFFERS
INTEGER NDBF  0 DETECTION BUFFER NUMBER
INTEGER NDBFHI  0 HIGHEST DETECTION BUFFER NUMBER TO USE
INTEGER KNDBUF(NMKOF)  0 COLOR/INTENSITY BUFFER
INTEGER NFRQ(NFRGSZ,NFRGCH)  0 LOCAL FREQUENCY TABLE (SCOPE INCLUDES
INTERNAL ROUTINE RESCOL)

INTEGER NFRCRO(10,15)  0 LOCAL CROSS FREQUENCY TABLE
INTEGER IPML  8 PPO LINE
INTEGER IPCMIN,IPCMAZ  8 MINIMUM AND MAXIMUM PPO COLUMN
INTEGER IPMLN,IPLMAX  8 MINIMUM AND MAXIMUM PPO LINE
REAL ADJLIN,ADJSAM  8 ADJUSTED LINE AND SAMPLE
INTEGER ML100,ML100M,ML100S  8 MSA LINE*100: LOW, HIGH, SPACING
INTEGER MLSLN  8 MSA LINE NUMBER
INTEGER MSASLO,MSASHI  8 LOW AND HIGH MSA SAMPLE
INTEGER LSTAT  8 I/O STATUS
INTEGER NPRLIN,NPRCOL  8 NUMBER OF PPO LINES AND COLUMNS
INTEGER LASTLN  8 LAST SCAN LINE READ

PROCEDURE

CALL TRACE

CALL OCTCK

DETERMINE HOW MANY BUFFERS TO USE

NDBFHI=MIN(NLIMCH,NH,NDBUF)
IF(NLIMCH,GT,NDBUF) CALL MNOTE1
= 'TOO MANY CHANNELS REQUESTED -- FIRST'.CBSMIN(NOBPNHM.2),
& 'CHANNELS PROCESSED')

C C
C INITIALIZE MINIMUM AND MAXIMUM PPO LINES AND COLUMNS
C
IPLMIN=PPO(1N.WMIN)
IPLMAX=PPO(1N.WMAX)
IPCIN=PPO(1N.WMIN)
IPCMAX=PPO(1N.WMAX)
C
C COMPUTE NUMBER OF PPO LINES AND COLUMNS
C
NPRLIN=IPLMAX-IPLMIN+1
NPRCOL=IPCMAX-IPCIN+1
C
C CLEAR LOCAL FREQUENCY TABLES
C
DO 250 I=1.NLINC
  DO 260 K=1.NFRQSZ
    NFREQ(K,1)=0
  250 CONTINUE
  260 CONTINUE
DO 280 I=1.10
  DO 290 K=1.15
    NFREQ(1,K)=0
  270 CONTINUE
  290 CONTINUE
C
C INITIALIZE LOW AND HIGH LINES AND SPACING
C
CALL AAP (AOJLIN.AOJSAM. FLOAT(IPLMIN),1.)
ML100L=AOJLIN+100.
CALL AAP (AOJLIN.AOJSAM. FLOAT(IPLMAX),1.)
ML100H=AOJLIN+100.
ML100S=MSAOWN(HLIN.WSP100)
C
C TURN ON COLOR SCREEN
C
CALL EAPRNT(O,I.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 AAP (AOJLIN.AOJSAM. FLOAT(IPLIN),FLOAT(IPCMIN))
  MSASLO=AOJSAM
  CALL AAP (AOJLIN.AOJSAM. FLOAT(IPLIN),FLOAT(IPCMAX))
  MSASHI=AOJSAM
  350  CONTINUE

L-308
IF(MSALIN.EQ.LASTLN) GO TO 320 IF ALREADY IN BUFFER(S)
DO 310 NDBF=1,NDBF+1
   CALL READ7N(MOTBUF(I,NDBF),(NWIDBF),ISTAT.
   MSALIN,LINCH(NDBF),MSASLO,MSASHI)
   IF(ISTAT.NE.'BADF') AND.
   (ISTAT.NE.'OFL') GO TO 310
   CALL HDFATL('CBS4CS(ISAT,1,4)',
   'WHILE READING DETECTION FILE')
   GO TO 900
310 CONTINUE
   LASTLN=MSALIN
320 CALL MSKPIX(MOTBUF(I,1),MOTBUF(I,1))
   IF((MOTBUF(PX81NT,1).EQ.'BYT').OR.
   (MOTBUF(PX81NT,1).EQ.'CHR').OR.
   (MOTBUF(PX81NT,1).EQ.'INT').OR.
   (MOTBUF(PX81NT,1).EQ.'NUL')) GO TO 330
   CALL HDFATL('INVALID BIN TYPE',
   'CBS4CS(MOTBUF(PX81NT,1,1,3))' IN PROP13)
   GO TO 900
330 IF((MOTBUF(PX81NT,1).EQ.'BYT') CALL RESCOL(KIBUF,(IPLIN),
   (IPMIN),(IPMAX),MOTBUF(NWIDBF),(NOBFU).GETOYT)
   IF((MOTBUF(PX81NT,1).EQ.'CHR') CALL RESCOL(KIBUF,(IPLIN),
   (IPMIN),(IPMAX),MOTBUF(NWIDBF),(NOBFU).GETCHR)
   IF((MOTBUF(PX81NT,1).EQ.'INT') CALL RESCOL(KIBUF,(IPLIN),
   (IPMIN),(IPMAX),MOTBUF(NWIDBF),(NOBFU).GETINT)
   IF((MOTBUF(PX81NT,1).EQ.'NUL') CALL RESCOL(KIBUF,(IPLIN),
   (IPMIN),(IPMAX),MOTBUF(NWIDBF),(NOBFU).GETNUL)
   CALL EAPRNT(KIBUF(KPNT1,KIBUF(KPNT2))
   IPLIN=IPLIN+1
350 CONTINUE
   C
   C MOVE DATA FROM LOCAL FREQ TABLES TO COMMON TABLE (REPLACING TICKS)
   C
   KTBLTY='FREQ'
   KTBLNM=NWIDBF
   DO 600 I=1,NL1MLH
   DO 550 K=1,NFRQSZ
   KFREQ(K,1)=NFREQ(K,1)
   550 CONTINUE
   600 CONTINUE
   DO 700 J=1,10
   DO 650 K=1,15
   KFRCRO(I,K)=NFRCRO(I,K)
   650 CONTINUE
   700 CONTINUE
   C
   C TURN OFF COLOR SCREEN AND RETURN -- NEXT CALL IS TO PROP19
   C
   CALL EAPRNT(0,1,KSOFF)
   CALL NVIATO(1,PR0129,PROP19)
   RETURN
   C
INTERNAL SUBROUTINE RESCOL1 0 RESAMPLE/SCREEN/COUNT FREQUENCY/COLOR

0 KIBUF. 0 COLOR/INTENSITY BUFFER
I IPLIN. 0 PPD LINE
I IPCMIN. 0 MINIMUM PPD COLUMN
I IPCMAX. 0 MAXIMUM PPD COLUMN
I KIBUF. 0 ARRAY OF MSS PIXEL BUFFERS
I NWBUF. 0 NUMBER OF WORDS IN ONE PIXEL BUFFER
I NOBUFS. 0 NUMBER OF PIXEL BUFFERS
I GETBIN 0 ROUTINE TO GET BIN VALUE--OETBYT.OETICE.OETINT.OETNUL

METHOD

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, COUNT FREQUENCY, AND SYMBOLIZE. INSERT TICKS.

EXTERNAL REFERENCES

AMP 0 ADJUSTED MSS COORD FOR PRINT/ PLOT COORD
OETICE 0 GET INTEGER-CHARACTER-EQUIVALENT FROM CHARACTER STRING
PUTBYT 0 PUT BYTE INTO BYTE STRING
MOVBYT 0 MOVE BYTE STRING

GLOBAL DECLARATIONS

INCLUDE KOMOWN.LIST 0 COMMON OUTPUT WINDOW PACKETS
INCLUDE KOMFREQ.LIST 0 COMMON FREQUENCY/TICK TABLE
INCLUDE KOMKLS.LIST 0 COMMON CLASSIFICATION INFO
INCLUDE PXBDEF.LIST 0 DEFINE PIXEL BUFFER STRUCTURE
INCLUDE KOMSYM.LIST 0 COMMON SYMBOL TABLE
INCLUDE KOMKLS.LIST 0 COMMON COLOR SCREEN PARAMETERS
INCLUDE WINDF.LIST 0 DEFINE WINDOW PACKETS
INCLUDE MAXINT.LIST 0 DEFINE MAXIMUM INTEGER

LOCAL DECLARATIONS

PARAMETER NUMBUF=8 0 NUMBER OF MPX BUFFERS
INTEGER NDDSBUF(NMIDSF,NDDSBUF) 0 ARGUMENT
INTEGER KIBUF(1) 0 ARGUMENT
INTEGER NDSBUF(NUMBUF) 0 BIN NUMBER OF SAMPLE 0 FOR EACH MPX BUFFER
INTEGER MS100L,MS100H,MS100S 0 MSA SAMPLE*100; LOW,HIGH,SPACING
INTEGER MSASAM 0 MSA SAMPLE NUMBER
INTEGER KIBIN 0 COLOR/INTENSITY BUFFER BIN POINTER
REAL ADJINSAM 0 ADJUSTED SAMPLE NUMBER
INTEGER IPIXL1, IPIXL2, IPIXL3, IPIXL4, IPIXL5, IPIXL6

& 8 PIXEL VALUE FOR EACH BUFFER
C
INTEGER IKENU 8 INTEGER-COLOR-EQUIVALENT (NEW)
INTEGER KIKE, KKENU 8 CODED INTEGER-COLOR-EQUIVALENT (CURRENT, NEW)
INTEGER KIE, KIENU 8 INTEGER-INTENSITY-EQUIVALENT (CURRENT, NEW)
INTEGER KIKE 8 CODED INTEGER-INTENSITY-EQUIVALENT (CURRENT)
C
C PROCEDURE
C
C INITIALIZE LOW AND HIGH SAMPLES AND SPACING
CALL ANP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCMIN))
MS100L = ADJSAM * 100.
CALL ANP (ADJLIN, ADJSAM, FLOAT(IPLIN), FLOAT(IPCMAX))
MS100H = ADJSAM * 100.
MS100S = MSADMN(WSAM, WSP100)
C
C INITIALIZE NUMBER OF BIN CONTAINING SAMPLE 0 FOR EACH MPX BUFFER
DO 150 NUMBUF = 1, NDBUF
    NBINSO(NUMBUF) = MDTBUF(PXLBIN, NUMBUF) - MDTBUF(PXLSAM, NUMBUF)
150 CONTINUE
C
C INITIALIZE PREAMBLE FOR COLOR-INTENSITY BUFFER
K1BUF(PXRECNI) = MDTBUF(PXRECNI)
K1BUF(PXLINI) = IPLIN
K1BUF(PXCHANI) = 0
K1BUF(PXQUALI) = 0
K1BUF(PXBINTI) = CBYTI
K1BUF(PXLBINI) = 1
K1BUF(PXCL0I) = IPCMIN
K1BUF(PXCH0I) = IPCMAX
K1BUF(PXNOINI) = 0
K1BUF(PXNOQDA) = 0
K1BUF(PXJ0I) = 0
K1BUF(PXJQ0I) = 0
C
C INITIALIZE BIN POINTER & FIRST BIN OF COLOR-INTENSITY BUFFER
KIBIN = K1BUF(PXLBINI)
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 INITIALIZE CURRENT INTEGER-INTENSITY-EQUIV & CODED INTEGER-COLOR-EQUIV
KIE = MAXINT
KIKE = MAXINT

L-311
C RESAMPLE/SCREEN RADIANC/E/COUNT FREQUENCY/LOOK UP SYMBOLS
C
DO 400 MS100=MS10OL,MS100H,MS100S
MSASAM=MS100/100
C BUFFER 1
C IF ((MSASAM.LT.MOTBUF(PXLSAM.1)).OR.
  (MSASAM.GT.MOTBUF(PXHSM.1))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPIXL1.
  HDTBUF(PXHINS.1),(MSASAM*NBINS0(1)))
  IPIXL2=IPIXL1  & SECOND CHANNEL VALUE SAME AS FIRST IF UNDEFINED
  IF (IPIXL1.GE.MOTBUF(PXNODA.1)) GO TO 350  & NO DATA
  IF (((IPIXL1.LT.LCVLO1).OR.
  (IPIXL1.GT.LCVHI1)) GO TO 360  & OUT OF RAO LIMITS
  IF (NLINCH-1.EQ.0) GO TO 320
C BUFFER 2
C IF ((MSASAM.LT.MOTBUF(PXLSAM.2)).OR.
  (MSASAM.GT.MOTBUF(PXHSM.2))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPIXL2.
  HDTBUF(PXHINS.2),(MSASAM*NBINS0(2)))
  IPIXL3=IPIXL2  & SECOND CHANNEL VALUE SAME AS FIRST IF UNDEFINED
  IF (IPIXL2.GE.MOTBUF(PXNODA.2)) GO TO 350  & NO DATA
  IF (((IPIXL2.LT.LCVLO(2)).OR.
  (IPIXL2.GT.LCVHI(2)) ) GO TO 360  & OUT OF RAO LIMITS
  IF (NLINCH-2.EQ.0) GO TO 290
C BUFFER 3
C IF ((MSASAM.LT.MOTBUF(PXLSAM.3)).OR.
  (MSASAM.GT.MOTBUF(PXHSM.3))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPIXL3.
  HDTBUF(PXHINS.3),(MSASAM*NBINS0(3)))
  IPIXL4=IPIXL3  & SECOND CHANNEL VALUE SAME AS FIRST IF UNDEFINED
  IF (IPIXL3.GE.MOTBUF(PXNODA.3)) GO TO 350  & NO DATA
  IF (((IPIXL3.LT.LCVLO(3)).OR.
  (IPIXL3.GT.LCVHI(3)) ) GO TO 360  & OUT OF RAO LIMITS
  IF (NLINCH-3.EQ.0) GO TO 260
C BUFFER 4
C IF ((MSASAM.LT.MOTBUF(PXLSAM.4)).OR.
  (MSASAM.GT.MOTBUF(PXHSM.4))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPIXL4.
  HDTBUF(PXHINS.4),(MSASAM*NBINS0(4)))
  IPIXL5=IPIXL4  & SECOND CHANNEL VALUE SAME AS FIRST IF UNDEFINED
  IF (IPIXL4.GE.MOTBUF(PXNODA.4)) GO TO 350  & NO DATA
  IF (((IPIXL4.LT.LCVLO(4)).OR.
  (IPIXL4.GT.LCVHI(4)) ) GO TO 360  & OUT OF RAO LIMITS
  IF (NLINCH-4.EQ.0) GO TO 230
C BUFFER 5
C IF ((MSASAM.LT.MOTBUF(PXLSAM.5)).OR.
  (MSASAM.GT.MOTBUF(PXHSM.5))) GO TO 350  & SAMPLE NOT IN BUFFER
  CALL GETBIN (IPIXL5.
  HDTBUF(PXHINS.5),(MSASAM*NBINS0(5)))
  IPIXL6=IPIXL5  & SECOND CHANNEL VALUE SAME AS FIRST IF UNDEFINED
  IF (IPIXL5.GE.MOTBUF(PXNODA.5)) GO TO 350  & NO DATA
  IF (((IPIXL5.LT.LCVLO(5)).OR.
  (IPIXL5.GT.LCVHI(5)) ) GO TO 360  & OUT OF RAO LIMITS
  IF (NLINCH-5.EQ.0) GO TO 210
IF (KINCH-S.EQ.0) GO TO 200

C BUFFER

IF (HSAASAN.LT.NOTBUF(PIXLSAM.B)) OR.
  HSAASAM.GT.NOTBUF(PIXLSAM.B)) GO TO 350  8 SAMPLE NOT IN BUFFER
  CALL GETBIN (PIXLS.)
  NOTBUF(PIXLSAM.B), HSAASAN=NBINS(B))
  IF ((IPIXLS.LT.LCVDG(8)) OR.
    (IPIXLS.GT.LCVDG(8))) GO TO 360  8 OUT OF RAD LIMITS

C COUNT FREQUENCY

           NFREQ(PIXLS+1,0)+NFREQ(PIXLS+1,1)
200       NFREQ(PIXLS+1,4)+NFREQ(PIXLS+1,5)
200       NFREQ(PIXLS+1,3)+NFREQ(PIXLS+1,4)
200       NFREQ(PIXLS+1,2)+NFREQ(PIXLS+1,3)
320       NFREQ(PIXLS+1,1)+NFREQ(PIXLS+1,1)

C INFO -- LOOK UP NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV

C CALL GETICE(IENU,
  KSYM(IPXLI+1,1), (5))
C CALL GETICE(IKENU,
  KSYM(IPXLZ+1,1), (6))
  GO TO 370

C NO DATA -- ASSIGN NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV

350 CALL GETICE(IENU,
  KSYM(ISYMN+1,1), (5))
C CALL GETICE(IKENU,
  KSYM(ISYMN+1,1), (6))
  GO TO 370

C NO INFO -- ASSIGN NEW INTEGER-INTENSITY-EQUIV & INTEGER-COLOR-EQUIV

360 CALL GETICE(IENU,
  KSYM(ISYMN+1,1), (5))
C CALL GETICE(IKENU,
  KSYM(ISYMN+1,1), (6))

C COUNT INTENSITY X COLOR CROSS FREQUENCY

370 NFRCRO(IENU+1,1KENU+1)=NFRCRO(IENU+1,1KENU+1)+1

C IF NEW CODED 1-K-E. PUT CODED 1-K-E & CODED 1-I-E IN BUFFER
C
  KIKENU=KSKIKE(IKENU+1)
  IF(KIKENU.EQ.KIKE) GO TO 360

L-313
M

DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

K11E=K11ENU
K1BIN=K1BIN+1
CALL PUTBYT(K1BUF(PXBINS).,K1BIN), K11E)
111E=111ENU
K11E=K11E+10
K1BIN=K1BIN+1
CALL PUTBYT(K1BUF(PXBINS).,K1BIN), K11E)
GO TO 400

C
C SAME COLOR -- IF SAME INTENSITY & BIN, UPDATE CODED 111E IN BUFFER
C
380 IF(111ENU.NE.111E) GO TO 380
IF(K11E.GT.1116) GO TO 380
K11E=K11E+10
CALL PUTBYT(K1BUF(PXBINS).,K1BIN), K11E)
GO TO 400

C
C SAME COLOR WITH NEW INTENSITY OR NEW BIN
C
390 111E=111ENU
K11E=K11E+10
K1BIN=K1BIN+1
CALL PUTBYT(K1BUF(PXBINS).,K1BIN), K11E)

C
C LOOP TO PROCESS NEXT PIXEL
C
400 CONTINUE

C
C STORE POINTER TO LAST BIN IN PREAMBLE & PAD BUFFER WITH 8 NUL'S
C
K1BUF(PXBINS)=K1BIN
CALL MOYBST(K1BUF(PXBINS).,K1BIN+1),(8),
0,(1),(1),(8)

C
C
900 RETURN
END

L-314
SUBROUTINE PROP19 (PICTURE DETECTION FILE(S) (PHASE 9))

HISTORY

E N SCHLOSSER LEC 05/17/79 REQUIREMENTS & DESIGN
E N SCHLOSSER LEC 11/06/79 STUBBED
E N SCHLOSSER LEMSCO 05/26/80 IMPLEMENTED

METHOD

CHECK DIAGNOSTIC COUNTERS AND PREPARE FOR NEXT PICTURE.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

NODATE NODLRH

EXCEPTIONS

NONE.

GLOBAL DECLARATIONS

INCLUDE KOMXOT.LIST COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMOWN.LIST COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDEF.LIST DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE NULCST.LIST DEFINE NULL CHARACTER STRING

EXTERNAL VIA TO
PR0000 NULLSUB

PROCEDURE

CALL TRACE

ON RETURN, CALL PR0000 TO GET COMMANDS

L-315
CALL NVIATO(PIR000, NULSUB)

DIAGNOSTICS???

IF(NOFATL.EQ.0) GO TO 980
   CALL NDNOTE( 'FATAL ERRORS -- NO PICTURE GENERATED')
   GO TO 980
IF(NDWARN.EQ.0) GO TO 980
   CALL NDNOTE( 'PREVIOUS WARNINGS -- NO PICTURE GENERATED')
   IF(BATCH.EQ.0) WRITE(6,995)
   FORMAT(IX, '***TRY AGAIN***')
   CALL NDCLRM(NULCST)
   GO TO 980

PRINT NWA OUTPUT WINDOW COORDINATES AND PREPARE FOR NEXT WINDOW

160 WRITE(6,965) MSAOHM(WLIN, WMIN)
165 FORMAT(IX, 'LINE ', IX)
   WRITE(6,975) MSAOHM(HSAM, WMIN), MSAOHM(HSAM, WMAX)
975 FORMAT(IX, 'SAMPLE ', IX, IX, 'SAMPLE ', IX)
   WRITE(6,995) MSAOHM(WLIN, WMAX)
995 FORMAT(IX)
   NWNOOW=NWNOOW+1

C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C
990 RETURN
END
SUBROUTINE PROXQT 0 INITIALIZATION ROUTINE FOR PRDDET

HISTORY

C
C
C
C	 E N SCHLOSSER LEC 11/89/79 ORIGINAL CODE
C	 E N SCHLOSSER LEC 01/23/79 ALLOW DEFAULT COMMANDS FROM MACDAN
C	 J C CRISP LEC 12/21/79 INITIALIZE LINCH & DELETE WINC
C	 J C CRISP LEMSCO 09/16/80 LINCH(1)=1, CHO PRDENS TO PRDDET

METHOD

C
C
C
C INITIALIZE PROGRAM, OPEN FILES, IDENTIFY SCENE, QUEUE DEFAULT COMMANDS.

MACHINE-DEPENDENT CODE

C
C
C
C UNIVAC EXEC-B PROGRAM FILE NAMING CONVENTIONS.

EXTERNAL REFERENCES

C
C
C
C NXTAT O NAME NEXT 'VIA' & 'TO' ROUTINES
C PSTART 0 PROGRAM START INITIALIZATION
C SYSAOO 0 ADD DISK SYMBOLIC FILE OR ELT TO SYSTIN RUNSTREAM
C OPLGEN 0 OPEN INPUT DETECTION FILES (E1, E2, ... )
C CLSHEO 0 PRINT CLASSIFICATION HEADING
C ODFATL 0 SUBMIT FATAL DIAGNOSTIC MESSAGE
C EXTERNAL PRODOD, NULSUB

EXCEPTIONS

C
C
C
C 1. MISSING DEFAULT COMMANDS GENERATE A FATAL DIAGNOSTIC.

GLOBAL DECLARATIONS

C
C
C
C INCLUDE KOKXOT.LIST 0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
C INCLUDE KOMKLS.LIST 0 COMMON CLASSIFICATION INFO

LOCAL DECLARATIONS

C
C
C
C INTEGER LOCFL1 0 LOCATION WITHIN DISK SYMBOLIC FILE ( IF > 0 )

L-517
C MAIN PROGRAMS/ROUNINES

C IDENTIFY PROGRAM
C CALL PSTART( 'DAM PRODSET(0000)' )
C ON RETURN, CALL PRO000 TO GET DEFAULT/USER COMMANDS
C CALL NVIATP (PR0000, NULSUB)
C OPEN DETECTION FILES AND IDENTIFY ERTS SCENE
C CALL OPNIEN
C CALL CLSHO01
C INITIALIZE LIMIT CHANNEL TO FIRST DETECTION CHANNEL
C NLINCH=1
C LINCII(1)=1
C QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE
C CALL SYSADD(LOCFIL. 'MACDANT.DEF-PRDSET.'')
C IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL. 'DANT.DEF-PRDSET.'')
C IF(LOCFIL.LE.0) CALL MDFATLI 'NO DEFAULT COMMANDS'
C RETURN TO NEXT STATEMENT IN CALLING ROUTINE
C RETURN
C END
SUBROUTINE IDCPRD  |  0 IDENTIFY CURRENT COMMAND SPECTS FOR PNTDET

UNIT 1 output unit

HISTORY

J C CRISP  LEC  12/26/70  REQUIREMENTS
J C CRISP  LEC  12/26/70  ALGORITHM DESIGN
J C CRISP  LEC  12/26/70  ALGORITHM CODING
J C CRISP  LEMSCO  05/28/80  CHANNEL TYPE IS 'DET'

METHOD

ENCODE FILE NAME FROM UNIT NUMBER. CHECK UNIT FOR 0 OR LEGAL
ALTERNATE PRINT FILE. BUILD CHARACTER BUFFER CONTAINING
CHANNEL NUMBER, RADIANCE LIMIT VALUES AND SYMBOLS, SPACING,
AND ORIGIN. OUTPUT BUFFER.

MACHINE-DEPENDENT CODE

UTILIZES UNIVAC EXEC 8 ER PRINTS AND ER PRNTAS

EXTERNAL REFERENCES

CBINIT  0  INITIALIZE CHARACTER BUFFER
CBCREST  0  CHARACTER BUFFER FOR CHARACTER STRING
CBVIN  0  CHARACTER BUFFER FOR INTEGER
CBVRL  0  CHARACTER BUFFER FOR REAL
CBVSTIN  0  CHARACTER STRING FOR INTEGER
ERPRNT  0  PRINT IMAGE ON TTY OR LINE PRINTER
ERPRTA  0  WRITE IMAGE TO ALTERNATE PRINT FILE
DOUBLE PRECISION CBVWIN  0  VARIABLE LENGTH CHAR STRING FOR INTEGER

EXCEPTIONS

1. OUTPUT WILL BE TRUNCATED TO FIT THE LENGTH OF THE OUTPUT BUFFER.
2. IF IUNIT IS NOT 0 OR A LEGAL ALTERNATE PRINT FILE (10-MALT-11),
   THEN A FATAL ERROR IS GENERATED.

GLOBAL DECLARATIONS

INCLUDE KOMXT.LIST  0  COMMON PROGRAM SWITCHES, COUNTERS
INCLUDE COMMON.LIST
INCLUDE COMMON.SYMBOL.TABLE
INCLUDE STRING.LIST
INCLUDE STRING.TABLE
INCLUDE STRING.WINDOW
INCLUDE STRING.WINDOW.PACKET
INCLUDE ICBUF1.LIST

LOCAL DECLARATIONS

INTEGER HAFIL (8)

PROCEDURE

CALL TRACE

CALL CST4IN (HAFIL(), UNIT)

CHECK FOR VALID OUTPUT UNIT/PRINT FILE

IF UNIT.EQ.81 OR
   AND (UNIT.LE.(10-MALTH-1)) GO TO 400;
   CALL NOPATL (COST4IN(UNIT), "", "BAD UNIT IN IDCPRO")
   GO TO 900

IDENTIFY CHANNEL NUMBER ;)

CALL CBINIT (ICBUFI)
CALL CBOST (ICBUFI, "CHAN.DEF")
DO 400 NLC=1,NLCH
   CALL CBOST (ICBUFI, ".")
   CALL CBWNT (ICBUFI, LINCH(NLC))
400 CONTINUE

IDENTIFY RADIANCE LIMIT VALUES AND SYMBOLS FOR FIRST CHANNEL

CALL CBOST (ICBUFI, ".")
CALL CBOST (ICBUFI, KLYTYP(1), (3))
CALL CBOST (ICBUFI, "")
CALL CBWNT (ICBUFI, LCVLOI)
CALL CBOST (ICBUFI, ".")
CALL CBOST (ICBUFI, KSYM(LCVLOI), (1))
CALL CBOST (ICBUFI, ".")
CALL CBWNT (ICBUFI, LCVHI)
CALL CBOST (ICBUFI, "")
CALL CBOST (ICBUFI, KSYM(LCVHI), (1))

L-320
C IDENTIFY RADIANCE LIMIT VALUES FOR ANY OTHER CHANNELS
C
IF (NLIMCN.LT.4) GO TO 600
DO 550 NLC=2,NLIMCN
   CALL CB4CST (ICBUFF1, '')
   CALL CB4IN ( '', LCVLO(NLC), 1 )
   CALL CB4CST (ICBUFF1, '')
   CALL CB4IN (ICBUFF1, LCVHI(NLC), 1 )
550 CONTINUE
C
C IDENTIFY SPACING
C
600 CALL CB4CST (ICBUFF1, '...SPA')
   DO 650 NAXIS=1,2
      CALL CB4CST (ICBUFF1, '')
      SPA=FLOAT(MSA0NN(NAXIS,WSP100))/100.
      IF (SPA.EQ.AINT(SPA)) CALL CB4IN (ICBUFF1, IFIX(SPA), 1 )
      IF (SPA.NE.AINT(SPA)) CALL CB4RL (ICBUFF1, SPA, 1, 4 )
   650 CONTINUE
C
C IDENTIFY ORIGIN
C
   CALL CB4CST (ICBUFF1, '...ORIG.SCAN')
   CALL CB4IN (ICBUFF1, MSA0NN(NLIM, WORIC), 1 )
   CALL CB4CST (ICBUFF1, '')
   CALL CB4IN (ICBUFF1, MSA0NN(WSAM, WORIC), 1 )
   CALL CB4CST (ICBUFF1, '')
C
C OUTPUT CHANNEL/RADIANCE/SPACING/ORIGIN INFO
C
   IF (UNIT.EQ.6) CALL ERPRNT (1.22,ICBUFF1)
   IF (UNIT.NE.6) CALL ERPRTA (NAMFILE,1.22,ICBUFF1)
C
C NORMAL RETURN
C
900 RETURN
END
SUBROUTINE OPRPRO 0 OPEN ALTERNATE PRINT FILES FOR PRDDET

HISTORY

E H SCHLOSSER  LEC  06/28/78  ORIGINAL CODE
J C CRISP     LEC  12/21/78  PRODEF FOR KPAGHI AND MALTHI

METHOD

A maximum of MALTHI (from PRODEF) alternate print files are
opened and initialized.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

OPENPR  OPEN ALTERNATE PRINT FILES

EXCEPTIONS

1. ONE PRINT FILE IS ALWAYS OPENED, UNLESS IN DATA/CHECKOUT MODE.
2. IF MALTHI IS GREATER THAN MALTHI (FROM PRODEF), IT WILL BE SET TO MALTHI.
3. IF KPAGHI IS GREATER THAN KPAGHI (FROM PRODEF), IT WILL BE SET TO KPAGHI.

GLOBAL DECLARATIONS

INCLUDE KOMXQT.LIST  2 COMMON PROGRAM EXECUTION SWITCHES.COUNTERS
INCLUDE PRODEF.LIST  2 PARAMETERS FOR PRDDET

LOCAL DECLARATIONS

NONE

PROCEDURE

L-322
C ----------------
C CALL TRACE
C
C NO MORE THAN MAITHI ALTERNATE PRINT FILES FOR PRTOET
C
MALTH-MAXO(MALTH.I)
MALTH-MIND(MALTH,MALTH)
C
C NO MORE THAN KPAGHI COLUMNS PER PAGE FOR PRTOET
C
KPAGH-MAXO(KPAGH.I)
KPAGH-MIND(KPAGH,KPAGH)
C
C OPEN FILE(S)
C IF(MDATAC.NE.0) GO TO 900 & DATA/CHECKOUT MODE
CALL OPENPR
C
C IDENTIFY ERTS SCENE
C
WRITE(10,225)
225 FORMAT(IX) & SKIP LINE
CALL CLSHOG(10)
C
900 RETURN
END
PROGRAM PRTCLASS

HISTORY

E M SCHLOSSER LEC 07/02/73 ORIGINAL CODE
MARY TOMPKINS LEC 01/05/80 UPGRADE DOCUMENTATION

METHOD

This program produces maps of classified ERTS MSS data on a computer line printer. The user specifies the scale, window dimensions, location, etc. of each map.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

NVIA TO NAME 'VIA' 'TO' ROUTINES
VIATO CALL 'VIA' 'TO' ROUTINES
VIA TO EXTERNAL PRC000, PRCXQT

EXCEPTIONS

1. This program is limited to that part of the world covered by the Clarke 1966 Spheroid (North America).

2. The program CLASSIFY must be executed before this program, preferably in the same run.

3. If PRCXQT does not call NVIA TO CHANGE THE 'VIA' AND/OR 'TO' ROUTINES. THEN PRTCLASS WILL CALL TO PRCXQT IN AN ENDLESS LOOP!

GLOBAL DECLARATIONS

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 KOMLEN.LIST COMMON I/O PKTS FOR DETECTION FILES (UNITS 2N
INCLUDE KOMINN.LIST COMMON INPUT WINDOW PACKETS
INCLUDE KOMONN.LIST COMMON OUTPUT WINDOW PACKETS

L-324
INCLUDE KOMNER.LIST  COMMON CRT$ SCENE PARAMETERS
INCLUDE KONKLS.LIST  COMMON CLASSIFICATION INFO
INCLUDE KONFIT.LIST  COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KONTBL.LIST  COMMON BLOCKS AND DEFINE PROCEDURES
INCLUDE KONDET.LIST  COMMON DETECTION FILE WINDOW PACKETS & DATES
INCLUDE KOMSYM.LIST  COMMON SYMBOL TABLE
INCLUDE KOMALT.LIST  COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS

C
C PROCEDURE

CALL NVIALEO( PRCO00,PRCXQT)  FIRST CALL IS VIA PRCO00 TO PRCXQT
100 CONTINUE
   CALL VIATE
   GO TO 100
END   STOP IS PERFORMED BY APPROPRIATE 'TO' ROUTINE
PRTCLASS HIERARCHY

PRTCLASS

- NVIATO
  - VIATO
- PRC000

  - PRCXQT
  - KMD...
  - PRCHAP
  - KMDXXX
  - PRCEX1

  - PSTART
  - OPN12N
  - SYSADD
  - OLET2N
  - CLOSPP
  - PSTOP
  - PABORT

DEF-PRTCLASS

  - OPRPRC
  - CALSYM
  - CALSCA
  - CALWIN
  - SUBMIN

  - OPENPR
  - MPRNRT
  - MPRNRT

  - READ2N
  - READ2N

L-326
PROGRAM PRTCLASS/VIRTUAL

HISTORY

E H SCHLOSSER LEC 08/02/74 ORIGINAL CODE
E H SCHLOSSER LEC 11/08/79 SNAP.FZEN (N O 'N' IN DEMAND)

METHOD

CONSTRUCT SNAP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPF5.
CONSTRUCT XQOT COMMAND TO EXECUTE REAL ABSOLUTE IN TPF5.
WRITE SNAP & XQOT 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 FIELD DATA 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 IONS 8 INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS 8 TERMINATE PROGRAM EXECUTION
DAM.PRTCLASS-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 $ASO-O & $PREP-O.

GLOBAL DECLARATIONS

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

XQOT OPTIONS ARE PRE-LOADED BY EXEC INTO REGISTER AS IN
(MASTER BIT NOTATION).

LOCAL DECLARATIONS

- AXRS
- SSLH, D-BANK
- LABSF, SSH FORM 6.6.18
- LABSDF, SSH 0501.1'F'.0 . LABEL. 1 WD. FORTRAN. FIELDATA
- LABING -SOFF--
- M APSDF, SSH 000.9.0.0 . DATA. 9 WDS. FIELDATA
- MAPING .8XQTS: MAP.FZN DAM.PRTCCLASS-MAP.PRTCCLASS .8XQTS'
- ADDSF, SSH 000.9.0.0
- ADDING .8XQTS: ADD DAM.SYS-MAPOPT .8XQTS'
- XQTSDF, SSH 000.9.0.0
- XQTING .8XQTS: XQT .1 PRTCCLASS .8XQTS'
- EOWSDF - 0 . END-OF-FILE STOP WORD
- PF FORM 12.6.18
- C S F A S O '.VASO.T 20 ..'
- C S F A D D '.SADD 20 ..'
- SAVREO RES 1
- I O P K T 1900 '.BS0'.WS 33.LABSDF.'0' 0

PROCEDURE

- S(01). 1-BANK
- PRTCCLASS* LA.U
- TNE.U A4.4 . SKIP NEXT INST IF A4<>4 (NOT DEMAND)
- SA.92 AD.ISAPING+2 . DEMAND! BLANK OUT IN OPTION
- LA AD.ISCSFSO) . ADDRESS OF BASO IMAGE
- ER CSFS . DO IT
- SA AD.ISAVREO) . STORE &
- PPRINT (PF 2.1.SAVREO) . PRINT BASO STATUS
- GETOPT . LOAD OPT LTRS INTO A2.A3.A4
- PUTOPT DS A2.XQTSING+2 . STORE OPTION LETTERS INTO XQTS IMAGE
- SA A4.XQTSING+4 . (3 WORDS -- MAX 18 OPT LETTERS)
- WRITE LA AD.ISIOPKTI) . ADDRESS OF I/O PACKET
- ER IOWS . WRITE SDF IMAGES TO 20.
- ADD LA AD.ISCSFADD) . ADDRESS OF SADD IMAGE
- ER CSFS . DO IT
- END PRTCCLASS

PRTCCLASS/VIRTUAL
PRTCCLASS-MAP
MAIN PROGRAM/RUTINES

PRTCCLASS OVERLAY STRUCTURE

HISTORY

E M. SCHLOSSER LEC 03/27/75 ORIGINAL CODE
E M. SCHLOSSER LEC 07/14/78 CHANGE OVERLAYS TO REDUCE THRASHERING
E M. SCHLOSSER LEC 01/31/79 MACRO COMMANDS & TIME COMMAND
MARY TOMPKINS LEMSCO 01/18/80 PEEK, POKE, .IF, .FI & OPTIMIZE OVERLAYS
MARY TOMPKINS LEMSCO 05/18/80 CHANNEL COMMAND

LIB DAM.

SEO S-MAIN
IN DAM.PRTCCLASS MAIN PROGRAM
IN DAM.NVIAO 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.NTAB/DAM DAM UNIT & TABLE GOES IN SAME SEO W/ FORTRAN I/O

UTILITIES FOR PHASE 0 COMMANDS --------------------------------------

SEO 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

SEO S-OPNCLPR*(S-MAIN)
IN DAM.OPRPRC OPEN ALT PRT FILE
IN DAM.CLOSPR CLOSE ALT PRT FILE

SEO S-DUMP*(S-MAIN)
IN DAM.OPMPTIC DUMP TIC TABLE
IN DAM.OMPWIN DUMP WINDOW PACKETS

PHASE 0 COMMANDS (FORTRAN I/O ALLOWED) -----------------------------

SEO START-STOP*(S-READS, S-OPNCLPR, S-DUMP)
IN DAM.PRCXOT PRTCCLASS INITIALIZATION ROUTINE
IN DAM.PRCEEI PRTCCLASS TERMINATION ROUTINE

SEO SPECIFY*, START-STOP
IN DAM.DETCHA GET/CHECK DETECTION CHANNEL NUMBER(S)
IN DAM.KNOCLE CLEAR WARNINGS/ERRORS
IN DAM.KNOCP OP GET/CHECK NUMBER OF OUTPUT COPIES
IN DAM.KNOCOU GET/CHECK COUNT PER PIXEL

L-329
IN DAN.KNOGEN . GET/CHECK DENSITY LIMITS
IN DAN.KNDEXP . EXPLAIN PROGRAM/COMMAND
IN DAN.KNDFI . END IF...FI BLOCK
IN DAN.KNOHEA . GET/CHECK PAGE HEADING(S)
IN DAN.KNOIF . BEGIN IF...FI BLOCK
IN DAN.KNODER . GET/CHECK TRANSVERSE MERIDIAN CENTRAL MERIDIAN
IN DAN.KNOKNOW . PRINT NEWS
IN DAN.KNOLEX . CONDITIONALLY PERFORM NEXT COMMAND
IN DAN.KNOOFF . TURN OFF MODE SWITCH(ES)
IN DAN.KNOON . TURN ON MODE SWITCH(ES)
IN DAN.KNOORI . GET/CHECK WINDOW ORIGIN
IN DAN.KNOPOP . SKIP TO TOP OF NEXT PAGE
IN DAN.KNOPS . POKE
IN DAN.KNOPS . POKE
IN DAN.KNPRF . GET/CHECK PRINTER SPECIFICATIONS
IN DAN.KNRAE . GET/CHECK RADIANCE LIMITS
IN DAN.KNRED . RENUMBER (GET/CHECK NEW WINDOW SEQUENCE NUMBER)
IN DAN.KNCSA . GET/CHECK WINDOW SCALE
IN DAN.KNCSY . GET/CHECK SYMBOLS
IN DAN.KNTIC . GET/CHECK TICK UNITS/INTERVALS
IN DAN.KNTIN . PRINT CLOCKS TIME & CHARGE TIME
IN DAN.KNOWIN . GET/CHECK WINDOW ENVELOPE/VERTICES
IN DAN.KNDXX . MACRO COMMANDS
IN DAN.KNDOZON . GET/CHECK UTM PROJECTION ZONE
IN DAN.KNOADD . DYNAMIC ADD
IN DAN.KNOSA . DYNAMIC SUB
IN DAN.KNOBR . DYNAMIC BRKPT
IN DAN.KNOFR . DYNAMIC FREE
IN DAN.KNOLO . DYNAMIC BLO

SEG MAPOUT*.START-STOP
IN DAN.PRCMAP . MAP RADIANCE/DENSITY/CLASS (PHASE 0)
SUBROUTINE PRC0001  0 CALL PHASE 0 SUBROUTINES FOR PRCTCLASS
0 NAME OF SUBROUTINE TO CALL (OR NULSUB)

HISTORY
--------
E H SCHLOSSER  LEC  03/27/77  ORIGINAL CODE
E H SCHLOSSER  LEC  06/27/78  DELETE RETN K'S & NUMERIC OPTIONS
E H SCHLOSSER  LEC  01/31/79  MACRO COMMANDS & COUNTER/RADIANCE/TIME
J C CRISP      LENUM  01/24/80  PEEK, POKE, IF, FI COMMANDS
MARY TOMPKINS  LENUM  05/18/80  CHANNEL COMMAND

METHOD  
--------
RETRIEVE NEXT COMMAND, VALIDATE IT, AND CALL ITS SUBROUTINE.

MACHINE-DEPENDENT CODE
-----------------------
NONE.

EXTERNAL REFERENCES
---------------------
READS      8 READ PUNCH CARD OR TERMINAL INPUT
GETBOL    0 GET ALPHABETIC COMMAND
INTEGER    6 INTEGER-CHAR-EQUIV FOR CHARACTER
WARNS      8 PRINT/LOG WARNING MESSAGE
PRC...     6 DEDICATED SUBROUTINE FOR COMMAND ... (SEE BELOW)
END...     6 COMMON SUBROUTINE FOR COMMAND ... (SEE BELOW)

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

GLOBAL DECLARATIONS
---------------------
INCLUDE NULLST.LIST  0 DEFINE NULL CHARACTER STRING

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

DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTES

C

INTEGER KOMD
INTEGER LSSTAT
INTEGER KASE

C FIRST 3 CHAR OF USER COMMAND (BLANK AFTER DONE
C READS STATUS ('COF' MEANS END-OF-FILE)
C MODIFIED 1-C-E OF FIRST CHAR OF COMMAND

C

PROCEDURE

C

CALL PREVIOUSLY NAMED SUBROUTINE

C

CALL TRACE
CALL NAMSUB  0 CALL TO NULSUB DOES NOTHING

C

CALL TRACE
CALL NAMSUB  0 CALL TO NULSUB DOES NOTHING

C

READ COMMAND FROM UNIT 9 (CARD READER OR TERMINAL)

C

KOMD='NUL'  0 IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
CALL READS(LSSTAT,"")  0 FILL BUFFER. BLANK CUE MESSAGE
IF(LSSTAT.NE."') KOMD='COFS'
IF(KOMD.NE."COFS") CALL GETSAL(KOMD,(3),NULSTAT)  0 GET 3 ALPHA CHARS

C

CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT

C

KASE=IC(E(KOMD)-ICE('a')+1  0 A TO Z = 1 TO 26

C

CASE STATEMENT ON MODIFIED 1-C-E OF COMMAND'S FIRST CHARACTER

C

IF(KASE.LT.11).OR.(KASE.GT.26) KASE=27  0 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

DETERMINE COMMAND. PERFORM COMMAND. CHANGE KOMD TO BLANK

C

*01 CONTINUE 8*** A
*02 CONTINUE 8*** B
00 TO 000

C

*03 CONTINUE 8*** C
IF(KOMD.EQ."CHA") CALL DE1(C,H,KOMD)  0 CHANNEL (DETECTION)
IF(KOMD.EQ."CLE") CALL KMODCL(KOMD)  0 CLEAR
IF(KOMD.EQ."COF") CALL KMODCOF(KOMD)  0 COPIES
IF(KOMD.EQ."COU") CALL KMODCOU(KOMD)  0 COUNT
1200 TO 000

C

*04 CONTINUE 8*** D
IF(KOMD.EQ."DEN") CALL KMODDEN(KOMD)  0 DENSITY
00 TO 000

C

L-333
400 CONTINUE 8*** E
   IF(KOMO.EQ."EGF") CALL PREEHE(KOMO)  0 END-OF-FILE CAUSES EXIT
   IF(KOMO.EQ."EX") CALL PREEER(KOMO)  0 EXIT
   IF(KOMO.EQ."EP") CALL KOMER(KOMO)  0 EXPLAIN
   GO TO 800

C 400 CONTINUE 8*** F
   IF(KOMO.EQ."FI") CALL KMOFPFI(KOMO)  0 FI (ENDIF)
   GO TO 800

C 407 CONTINUE 8*** G
   GO TO 800

C 408 CONTINUE 8*** H
   IF(KOMO.EQ."HEA") CALL KMDHEA(KOMO)  0 HEADING
   GO TO 800

C 409 CONTINUE 8*** I
   IF(KOMO.EQ."IF") CALL KMDIF(KOMO)  0 IF
   GO TO 800

C 410 CONTINUE 8*** J
411 CONTINUE 8*** K
412 CONTINUE 8*** L
   GO TO 800

C 413 CONTINUE 8*** M
   IF(KOMO.EQ."MAP") CALL PREMAP(KOMO)  0 MAP
   IF(KOMO.EQ."HER") CALL KMDHER(KOMO)  0 MERIDIAN
   GO TO 800

C 414 CONTINUE 8*** N
   IF(KOMO.EQ."NEW") CALL KMDNEW(KOMO)  0 NEWS
   IF(KOMO.EQ."NEX") CALL KMDNEX(KOMO)  0 NEXT
   GO TO 800

C 415 CONTINUE 8*** O
   IF(KOMO.EQ."OFF") CALL KMODFF(KOMO)  0 OFF
   IF(KOMO.EQ."ON") CALL KMODON(KOMO)  0 ON
   IF(KOMO.EQ."ORI") CALL KMODORI(KOMO)  0 ORIGIN
   GO TO 800

C 416 CONTINUE 8*** P
   IF(KOMO.EQ."PAO") CALL KMDPAO(KOMO)  0 PAGE
   IF(KOMO.EQ."PEE") CALL KMDPEE(KOMO)  0 PEEK
   IF(KOMO.EQ."POK") CALL KMDPOK(KOMO)  0 POKE
   IF(KOMO.EQ."PRI") CALL KMDPRI(KOMO)  0 PRINT
   GO TO 800

C 417 CONTINUE 8*** Q
   GO TO 800

C 418 CONTINUE 8*** R
   IF(KOMO.EQ."RAD") CALL KMDRAD(KOMO)  0 RADIANCE
   IF(KOMO.EQ."REN") CALL KMDREN(KOMO)  0 RENUMBER
   GO TO 800
\begin{verbatim}
C 419 CONTINUE 8*** S
   IF(KOMD.EQ.'SCA') CALL KMDSCA(KOMD) 8 SCALE
   IF(KOMD.EQ.'SYM') CALL KMDSYM(KOMD) 8 SYMBOLS
   GO TO 800
C 420 CONTINUE 8*** T
   IF(KOMD.EQ.'TIC') CALL KMDTIC(KOMD) 8 TICKS
   IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD) 8 TIME
   GO TO 800
C 421 CONTINUE 8*** U
422 CONTINUE 8*** V
   GO TO 800
C 423 CONTINUE 8*** W
   IF(KOMD.EQ.'WIN') CALL KMDWIN(KOMD) 8 WINDOW
   GO TO 800
C 424 CONTINUE 8*** X
425 CONTINUE 8*** Y
   GO TO 800
C 426 CONTINUE 8*** Z
   IF(KOMD.EQ.'ZON') CALL KMDZON(KOMD) 8 ZONE
   GO TO 800
C 427 CONTINUE 8*** NOT ALPHABETIC
   IF(KOMD.EQ.'SAD') CALL KMDSAD(KOMD) 8 SADD
   IF(KOMD.EQ.'SAS') CALL KMDSAS(KOMD) 8 SASG
   IF(KOMD.EQ.'SBR') CALL KMDSBR(KOMD) 8 SBKP
   IF(KOMD.EQ.'SFR') CALL KMDSFR(KOMD) 8 SFREE
   IF(KOMD.EQ.'SLO') CALL KMDSLO(KOMD) 8 SLOG
C C IF COMMAND WAS NOT FOUND, TRY MACRO-COMMAND
   800 IF(KOMD.NE.' ') KOMD='PRC-' 8 1ST 3 CHAR OF PROG NAME PLUS '-'
   IF(KOMD.NE.' ') CALL KMDXXX(KOMD) 8 MACRO COMMAND HANDLER
C C COMMAND IS INVALID IF STILL NOT FOUND
   IF(KOMD.NE.' ') CALL WARNS( 'INVALID COMMAND --'
C C FORCE ALL FORTRAN I/O ROUTINES INTO SAME SEQ AS PRG000 (NEVER PERFORMED)
   IF(KOMD.EQ.'JUNK') READ(895,895) KOMD
   895 FORMAT(IX)
C C RETURN TO MAIN FOR CALL VIA/T0 NAMED SUBROUTINE IN ANY OVERLAY
   RETURN
\end{verbatim}
END
SUBROUTINE PXXEIU (A TERMINATION ROUTINE FOR PRCLASS)
U CMD): 8 I: FIRST 3 CHARMS OF COMMAND: O: SPACES

C (E N SCHLOSSER)

INCLUDE KOMXQT.LIST
INCLUDE NULLCST.LIST 8 DEFINE NULL CHARACTER STRING
CALL TRACE

WRITE(*,125)
125 FORMAT('**PROGRAM TERMINATION**')
IF(MDATA.EQ.0) CALL CLOSPR
IF(MDATA.NE.0) CALL PAORT( NULLCST).
WRITE(*,165)
165 FORMAT('DO YOU WANT DETECTION FILES SAVED?')
CALL READSLSTAT.
IF(SAVE.EQ.'N')
CALL GETSKH(SAVE.1), NULLCST)

DELETE DETECTION FILES
IF(SAVE.NE.'Y') CALL DEL2N

TERMINATE PROGRAM
CALL PSTOP(NULLCST)

PSTOP DOES NOT RETURN
END
SUBROUTINE PRCMAP

A INITIATE PRINTING OF CLASSIFICATION MAP(S)

U KONG

1: FIRST 3 CHARs OF COMMAND: SPACES

HISTORY

E M SCHLOSSER  LEC  07/22/73  ORIGINAL CODE
E M SCHLOSSER  LEC  06/20/79  DELETE RETN K & ADD OPRPRC
E M SCHLOSSER  LEC  03/16/79  CHG SUBWIN FROM INTERNAL TO EXTERNAL
E M SCHLOSSER  LEMSCO 05/16/80  CHECK NUMBER OF CHANNELS REQUESTED

METHOD

GET/CHECK/CALIBRATE SPECS. OPEN PRINT FILE(S) IF NOT ALREADY OPEN. THEN
GENERATE WINDOW OR SUBWINDOW MAPS.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSIN  GET INTEGER DATA FIELD FROM UNIT S
WARNS  GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT S
MDMARN  PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
CALSYM  CALIBRATE SYMBOL TABLE FOR OVERPRINTING
CALSCE  CALIBRATE TRANSFORMATION COEFFICIENTS FOR SCALE
CALWIN  CALIBRATE WINDOW ENVELOPES
OPRPRC  OPEN ALTERNATE PRINT FILE(S) FOR PRTCLASS
SUBWIN  GENERATE SUBWINDOW MAPS
MAPRNT  GENERATE WINDOW MAPS
MDCLRWH  CLEAR COUNT OF 'WARNING' DIAGNOSTICS
EXTERNAL MAPRNT

EXCEPTIONS

1. 'MAP' MAY NOT BE A DEFAULT COMMAND.
2. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE MAP.

GLOBAL DECLARATIONS

INCLUDE KOMXQT.LIST  8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST  8 COMMON CLASSIFICATION INFO
INCLUDE WINDEF.LIST  # DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMINW.LIST  # COMMON INPUT WINDOW PACKETS
INCLUDE KOMOUT.WLIST # COMMON OUTPUT WINDOW PACKETS
INCLUDE NULCST.LIST  # DEFINE NULL CHARACTER STRING

**LOCAL DECLARATIONS**

**LOCAL DECLARATIONS**

INTEGER NSUBW  # NUMBER OF SUB-WINDOWS REQUESTED
INTEGER INTEMP  # TEMPORARY

**PROCEDURE**

**CALL TRACE**

**GET/CHECK NUMBER OF SUB-WINDOWS**

NSUBW=0
CALL GETSUB(NSUBW, 1.500, 'BAD NUMBER OF SUB-WINDOWS -- ')
IF(MCONF. NE.0) WRITE(6,115) NSUBW
115 FORMAT(' MAP. '13.' SUB-WINDOWS')

**CALIBRATE/CHECK SPECIFICATIONS**

IF(NWINDOW.EQ.0) CALL WARN1('INVALID DEFAULT COMMAND -- ')
CALL GETSUB(INTEMP, '1.1, 'EXTRA MAP SPECIFICATION -- ')
IF(INCH.GT.1) CALL MNOTE(' MORE THAN 1 CHANNEL REQUESTED -- ONLY FIRST CHANNEL PROCESSED ')
CALL CALSYM
CALL CALSCA(FLOAT(LINCH,FLOAT(KINCH1)
CALL CALWIN2.1
IF(MCHECK. NE.0) GO TO 900

**CHECK IF ANY DATA LIES WITHIN OUTPUT WINDOW**

IF((MSOWW(WLIN.WMIN).GE.MSAWW(WLIN.WMAX)) .OR. 
(MSAWW(WSAM.WMIN).LE.MSAWW(WSAM.WMAX)).OR. 
(MSAOWW(WSAM.WMIN).GE.MSAWW(WSAM.WMAX)).OR. 
(MSAOWW(WSAM.WMAX).LE.MSAWW(WSAM.WMIN))). CALL MWARN('NO DATA WITHIN WINDOW')

**OPEN PRINT FILE(S) IF NOT OPEN, CLEAR WINDOW NUMBER & RESET PAGE NUMBER**

IF(NDTOTL. NE.0) GO TO 900
IF(NWINDOW.LT.01) CALL OPRPRC ' OPEN ALT PRT FILE(S) BEFORE 1ST WINDOW
NWINDOW=ABS(NWINDOW)
NPAGE=0

L-339
C GENERATE WINDOW Map
C
IF(NSUBH.NE.0) GO TO 400
CALL MAPRNT
GO TO 900
C
C GENERATE GEOGRAPHIC SUBWINDOW MAPS
C
400 IF((KS0WM(WTIC).NE."DEG"),AND.
& (KS0WM(WTIC).NE."MIN") GO TO 800  & NOT GEOGRAPHIC
CALL SUBWIN(GEOWWN.NSUEH.MAPRNT)
GO TO 900
C
C GENERATE UTM SUBWINDOW MAPS
C
600 IF((KS0WM(WTIC).NE."KM"),AND.
& (KS0WM(WTIC).NE."MET") GO TO 800  & NOT UTM
CALL SUBWIN(UTM0WWN.NSUEH.MAPRNT)
GO TO 900
C
C INVALID TICK INTERVAL FOR GENERATING SUBWINDOW MAPS
C
800 CALL M0WARN( 'SUBWINDOWs NOT ALLOWED FOR CURRENT PRIMARY TICKS')
C
C ANY DIAGNOSTICS???
C
900 IF(NDFL.T).EQ.0) GO TO 920
CALL MONOTE( 'FATAL ERRORS -- NO MAP GENERATED')
GO TO 990
920 IF(N00M3R.EQ.0) GO TO 990
CALL MONOTE( 'PREVIOUS WARNINGS -- NO MAP GENERATED')
IF(MBATCH.EQ.0) WRITE(6,925)
925 FORMAT(4X,'**TRY AGAIN!**')
CALL MOCLRM(NULCST)
990 WRITE(6,995)
995 FORMAT(2X)
RETURN
END
SUBROUTINE PRCXOT & INITIALIZATION ROUTINE FOR PRTCLASS

HISTORY
---
E H SCHLOSSER  LEC  09/18/74  ORIGINAL CODE
E H SCHLOSSER  LEC  01/31/79  ALLOW DEFAULT COMMANDS FROM MACDAM
E H SCHLOSSER  LEMSIO  05/16/80  LINECH(1)=1

METHOD
------
INITIALIZE PROGRAM, OPEN FILES, IDENTIFY SCENE, QUEUE DEFAULT COMMANDS.

MACHINE-DEPENDENT CODE
-----------------------
UNIVAC EXEC-9 PROGRAM FILE NAMING CONVENTIONS.

EXTERNAL REFERENCES
-------------------
NVIAIO  NAME NEXT 'VIA' & 'TO' SUBROUTINES
PSTART  PROGRAM START INITIALIZATION
OPN2N   OPEN INPUT DETECTION FILES (21, 22, ...)
CLSHOO  PRINT CLASSIFICATION HEADING
SYSSAO  ADD DISK SYMBOLIC FILE OR ELT TO SYSIN RUNSTREAM
MFATL   SUBMIT FATAL DIAGNOSTIC MESSAGE

EXCEPTIONS
----------
1. MISSING DEFAULT COMMANDS GENERATE A FATAL DIAGNOSTIC.

GLOBAL DECLARATIONS
----------------------
INCLUDE KOMXOT.LIST  COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST  COMMON CLASSIFICATION INFO
EXTERNAL PRCOO00.NULSUB

LOCAL DECLARATIONS
-------------------
INTEGER LOCFLF  LOCATION WITHIN DISK SYMBOLIC FILE ( IF > 0 )

PROCEDURE
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

---------

IDENTIFY PROGRAM

CALL PSTART('DAM PRVCCLASS(809)')

ON RETURN, CALL PRC000 TO SET DEFAUL/TUSER COMMANDS

CALL NIATOS( PRCD000.NULSUB)

OPEN DETECTION FILE(S) AND IDENTIFY ERROR SCENE

CALL OPHIEN
CALL CLSH00(6)

INITIALIZE LIMIT CHANNEL TO FIRST DETECTION CHANNEL

NLINCH=1
LINCH(1)=1

QUEUE DEFAUL/T COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE

CALL SYSADD(LOCFIL, 'MACDAM', 'DEF-PRTCCLASS', '')
IF(LOCFIL.LE.0) CALL SYSADD(LOCFIL, 'DAM', 'DEF-PRTCCLASS', '')
IF(LOCFIL.LE.0) CALL MFAIL('NO DEFAUL/T COMMANDS')

RETURN TO NEXT STATEMENT IN CALLING ROUTINE

RETURN

END
SUBROUTINE MAPRNT & MAP DETECTION FILE DATA FOR PRTCCLASS

HISTORY

E H SCHLOSSER      LEC  07/22/73   ORIGINAL CODE
E H SCHLOSSER      LEC  03/19/79   'DEN'/‘RAD'/‘CLA' DETECTION FILES
D A BECK           LEC  12/30/79   REVISE FOR PX80FX FORMAT BUFFERS
J C CRISP          LEMSCO 05/16/80   READ CHANNEL LIMIT, INSTEAD OF 1
J C CRISP          LEMSCO 08/18/80   ADD 4 WORDS TO PRINT BUFFER

METHOD

GENERATE TICK TABLE IN MEMORY AND TABULAR DATA ON MAP UNIT 0.
BREAK THE PRINT WINDOW INTO SECTION(S) MALT UNIT(S) BY
(KPAGB) COLUMNS WIDE. PROCESS A SECTION OF WINDOW AT A TIME.
WITHIN EACH SECTION, READ/MASK/SCREEN/RESAMPLE DETECTION SCAN LINE(S)
TO PRINT LINES AND SYMBOLIZE/TICK/OUTPUT PRINT LINES.

MACHINE-DEPENDENT CODE

ASSUMES 6 CHARS PER INTEGER BIN.
ARGUMENTS IN CALLS TO ROUTINES UTILIZING EXEC-BER ROUTINES MAY
BE MACHINE-DEPENDENT.

EXTERNAL REFERENCES

A%P        5 ADJUSTED FROM PPD COORD
P%4        5 PPD COORD FROM ADJUSTED
P%THOD      5 PRINT MALT UNIT HEADING
P%SYML      5 PRINT SYMBOL LEGEND
O%ENTIC     5 GENERATE TICK TABLE
P%ADN      5 READ DETECTION FILE(S)
M%KPIX      5 MASK PIXELS EXTERIOR TO WINDOW
P%ESPRC      5 RESAMPLE DETECTION PIXELS
P%PROFI     5 PRINT/OVERPRINT FILE(S)
M%DFAIL      5 '/PRINT/COUNT/LOG 'FATAL ERROR' MESSAGES
M%DWARN      5 '/PRINT/COUNT/LOG 'WARNING' MESSAGES
P%PRTA      5 PRINT MESSAGE ON ALTERNATE PRINT FILE

DOUBLE PRECISION CSS%CS 5 VARIABLE LENGTH CST FOR FIXED LENGTH CST
TO
EXTERNAL PESPRC, OETINT, OETBYT, GETICE, GETNUL

EXCEPTIONS

1. IF THE NUMBER OF COLUMNS TO BE PRINTED IS GREATER THAN
THE NUMBER OF COLUMNS PER PAGE TIMES MALTHI THEN
GENERATE A WARNING.

2. IF THE COUNTS PER PIXEL IS < ZERO OR > 20 THEN
GENERATE A WARNING.

3. IF THE ORIGIN TYPE IS NOT 'SCA' OR 'DEG' OR 'MIN' OR
'KH' OR 'NET' THEN GENERATE A FATAL ERROR.

4. IF THE BIN TYPE OR THE DETECTION BUFFER IS NOT 'INT' OR 'BYT'
OR 'CHR' THEN GENERATE A FATAL ERROR.

5. ANY WARNING OR FATAL ERROR PREVENTS GENERATION OF THE MAP.

6. THE FOLLOWING VALUES OF ISTAT PRODUCE THE FOLLOWING RESULTS

<table>
<thead>
<tr>
<th>CONDITION</th>
<th>NEAR LINE</th>
<th>PRINT CELL</th>
<th>DIAGNOSTIC</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>'COF'</td>
<td>':'</td>
<td>NO DATA(':')</td>
<td>NONE</td>
<td>PRINT LINE</td>
</tr>
<tr>
<td>'BADR'</td>
<td>'?'</td>
<td>NO DATA(':')</td>
<td>NONE</td>
<td>PRINT LINE</td>
</tr>
<tr>
<td>'BADF'</td>
<td>N/A</td>
<td>N/A</td>
<td>FATAL</td>
<td>RETURN</td>
</tr>
<tr>
<td>'OFL'</td>
<td>N/A</td>
<td>N/A</td>
<td>FATAL</td>
<td>RETURN</td>
</tr>
</tbody>
</table>

7. LOCAL DECLARATIONS

<table>
<thead>
<tr>
<th>DECLARATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>INCLUDE KOMKLOT.LIST</td>
</tr>
<tr>
<td>INCLUDE KOMKLO.LIST</td>
</tr>
<tr>
<td>INCLUDE WINDIF.LIST</td>
</tr>
<tr>
<td>INCLUDE KONDEF.LIST</td>
</tr>
<tr>
<td>INCLUDE KOMWIN.LIST</td>
</tr>
<tr>
<td>INCLUDE KOMALT.LIST</td>
</tr>
<tr>
<td>INCLUDE KOMSYN.LIST</td>
</tr>
<tr>
<td>INCLUDE KOMTBL.LIST</td>
</tr>
<tr>
<td>INCLUDE PXBDEF.LIST</td>
</tr>
<tr>
<td>INCLUDE PXDEF.LIST</td>
</tr>
<tr>
<td>INTEGER IN DET BUF</td>
</tr>
<tr>
<td>PARAMETER NWPDFF</td>
</tr>
<tr>
<td>WDS PRF BUF = WDS PREAM+(MAX FILES*MAX COLS/P4-MEATLN COLS-MARG COLS)-1</td>
</tr>
<tr>
<td>PARAMETER NWPDF = (N48+1)+MALTHI*KPAGHI-2</td>
</tr>
<tr>
<td>INTEGER IDETDF(NWPDFF)</td>
</tr>
<tr>
<td>INTEGER IMPDFN(IKPDFF)</td>
</tr>
<tr>
<td>INTEGER IPXMIN,IPXMAX</td>
</tr>
<tr>
<td>INTEGER IPCMIN,IPCMAX</td>
</tr>
<tr>
<td>INTEGER IPCGD,IPCHI</td>
</tr>
<tr>
<td>INTEGER IPCINC</td>
</tr>
</tbody>
</table>
INTEGER NUMIT 0 UNIT NUMBER OF ALTERNATE PRINT FILE
INTEGER HSALIN 0 CURRENT MSA LINE NUMBER
INTEGER IPLIN 0 CURRENT PRINT LINE NUMBER
INTEGER HSASLO,HSASHI 0 LOW, HIGH MSA SAMPLE NUMBER IN SECTION
INTEGER ISTAT 0 I/O STATUS RETURN WORD
INTEGER IPLNXS 0 PRINT LINE NUMBER FOR NEXT SCAN LINE
REAL ADJLIN,ADJSAM 0 ADJUSTED LINE, SAMPLE
REAL PPOLIN,PPOCOL 0 PPO LINE, COLUMN
INTEGER NTLCHR 0 NEAT LINE CHARACTER
INTEGER MROLS.MRORS 0 MARGIN LEFT, RIGHT CHAR STRING

C

C DECLARATIONS WHICH ARE GLOBAL TO MAPRNT'S INTERNAL SUBROUTINES
C
PARAMETER NSAVMX=60 0 MAXIMUM * OF SAVE LOCATIONS IN ISAVSY
INTEGER ISAVSY(NSAVMX,2) 0 SAVE TBL OF BIN*/SYMBOLS REPLACED BY TICKS/MALO
INTEGER HSVSAVED 0 # OF SYMBOL(S) REPLACED BY TICK(S) & SAVED
INTEGER IPLTIC,IPCSTIC,JSYTIC 0 TICK LINE, COLUMN, AND SYMBOL
INTEGER JPSTIC 0 PRINT SPEC FOR PROVFI

C

C PROCEDURE
C

CALL TRACE

C

C INITIALIZE MINIMUM AND MAXIMUM PRINT LINES AND COLUMNS IN WINDOW
C
IPLMIN=PPDOWN(MLIN,WRMIN)
IPLMAX=PPDOWN(MLIN,WRMAX)
IPCHMIN=PPDOWN(MCOL,WRMIN)
IPCHMAX=PPDOWN(MCOL,WRMAX)

C

C BREAK WINDOW INTO UNITS, EACH NOT MORE THAN KPAGE-6 COLUMNS WIDE
C
NUNITMAX=(IPCHMAX-IPCHMIN)/(KPAGE-6)
IF(NUNITMAX.OT.WMAX) CALL MDWARN('WINDOW TOO WIDE')

C

C CHECK FOR VALID MAXIMUM COUNT PER PIXEL
C
IF(KTIPX.LT.0).OR.(KTIPX.GT.20) CALL MDWARN('INVALID COUNT PER PIXEL')

C

C CHECK FOR DIAGNOSTICS
C
IF(NOTOL.NE.0) GO TO 999

C

C CHECK FOR VALID ORIGIN TYPE
C
IF(KSYOWH(MORIG).EQ.'SCA') GO TO 100
IF(KSYOWH(MORIG).EQ.'DEG') OR.(KSYOWH(MORIG).EQ.'MIN') GO TO 120

L-345
IF(KSTWNN(HROR10).EQ."KM") .OR. (KSTWNN(HROR10).EQ."ME") GO TO 140
CALL HDFATL( "INVALID ORIGIN TYPE IN MAPRNT")
GO TO 300

C
C  PRINT WINDOW NUMBER AND ORIGIN
C
100 WRITE(6,100) NNTNDN,MSAONN(HLIN,HROI0),MSAONN(HSAM,HROI0)
100 FORMAT(6,'WINDOW *','13.'(ORIGIN ',','14.' LINE ',','14.' SAMPLE')')
GO TO 150
120 WRITE(6,120) NNTNDN,ODDNN(MLAT,HROR10),ODDNN(HLON,HROR10)
120 FORMAT(6,'WINDOW *','13.'(ORIGIN ','FD,4.' LAT ','FD,4.' LON')')
GO TO 150
140 WRITE(6,140) NNTNDN,UTMNN(NEA,HROR10),UTMNN(HNOO,HROR10)
140 FORMAT(6,'WINDOW *','13.'
150 (ORIGIN ',','-3P,FD,3.' KH E ','FD,3.' KH N')')

C
C  GENERATE TABULAR DATA (UNIT 0)
C
150 NUNIT=0
150 IF(LEGENF.EQ.0) GO TO 200 8 NO UNIT 0 UNLESS LEGEND MODE SWITCH IS ON
NITLO=0
NITHI=0
INCLUDE NATROT.LIST
NIT=0
NUNIT=NTAB(NIT)
CALL NATNDDO(NUNIT) 8 PRINT UNIT HEADING
CALL PRTSYML(NUNIT) 8 PRINT SYMBOL LEGEND
150 IF(KTIPIX.LE.1).AND.(KTIPIX.LE.9)
6 WRITE(NUNIT,170) KTIPIX
170 FORMAT(6(1 COUNT = 1/",11. PXE0.' PIXE0/'))
170 IF(KTIPIX.LE.9)
6 WRITE(NUNIT,190) KTIPIX
190 FORMAT(6(1 COUNT = 1/",12. PXE0.' PIXE0/'))
200 CALL SENTIC(NUNIT) 8 GENERATE TICK TABLE AND PRINT ON NUNIT IF NOT 0

C
C  INITIALIZE LOW/HIGH PRINT COLUMNS FOR FIRST SECTION OF WINDOW
C
1PCINC=(KPAGE-8)*MALTH
1PCLN=IPCHN
IPCHN=HDHI(IPCLO+1PCINC-11.IPCHN)

C
C  PROCESS WINDOW IN SECTIONS. EACH NOT MORE THAN MALTH PRINT UNITS WIDE
C
DO 800 NITLO=1,NITHI,MALTH 8 LOW UNIT FOR EACH SECTION OF WINDOW
NITHI=MIN((NITLO-MALTH-11,NITHI) 8 HIGH UNIT
INCLUDE NATROT.LIST 8 ROTATE THE ASSIGN OF LOGICAL UNITS TO MAP UNITS

C
C  INITIALIZE FIRST TICK, EXACT FIRST PPD LINE#, APPROMATE FIRST SCAN LINE#
C
CALL SENTIC(IPLTIC1,IPCTIC,JSTIC)
IPLHN=IPLHIN
G^ OAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

CALL ANPIADJLIN.ADJSAM. O INSURE THAT MSALIN
FLOAT(IPLMIN-1),0.1 O = FIRST SCAN LINE
MSALIN=ADJLIN 0 WHICH MAPS TO IPLMIN

C

C PRINT UNIT HEADING(S)
C
DO 300 NIT=NITLO,NITHI 0 COMPUTE LOGICAL UNIT FOR MAP UNIT
NUNIT=NTABInit(NUNIT)
300 CONTINUE
C
C GENERATE AND PRINT TOP COLUMN SCALE/BORDER FOR UNIT(S) IN CURRENT SECTION
C
CALL SANSCL(IPRTBF,(IPLMIN),(IPCLO),(IPCH1))
NUNIT=NBINIT(NIT)
IPRTBF=IPXRECH1-MNDND*1000
CALL PROVF1(NUNIT, '##9##', '##9##', '##9##', '##9##', IPRTBF)

C
C PROCESS ALL PRINT LINES IN CURRENT SECTION OF WINDOW
C
400 CONTINUE 0 DO 940 UNTIL IPLIN > IPLMAX
C
C COMPUTE LOW AND HIGH DETECTION SAMPLES FOR CURRENT PRINT LINE
C
CALL ANPIADJLIN.ADJSAM. FLOAT(IPLIN),FLOAT(IPCLO)
MSASLO=ADJSAM 0 MUST BE (= NFA SAMPLE WHICH MAPS TO IPCLO
CALL ANPIADJLIN.ADJSAM. FLOAT(IPLIN),FLOAT(IPCLO)
MSASHI=ADJSAM+1.0 0 MUST BE >= NFA SAMPLE WHICH MAPS TO IPCH1
CALL PROB1(IPRTBF,(IPLIN),(IPCLO),(IPCH1))

C
C READ/MASK/SCREEN/RESAMPLE ANY SCAN LINE(S) WHICH MAP TO CURRENT PRINT LINE
C
410 CONTINUE 0 DO 480 WHILE IPLNXS <= IPLIN
CALL PW41PPOOLN,PPCOL,FLOAT(MSALIN),0.1
IPLNXS=PPOLIN
IF(IPLNXS.GT.IPLIN) GO TO 500 0 NO MORE NFA LINES FOR IPLIN
IF(IPLNXS.LT.IPLIN) GO TO 480 0 NOT YET NFA LINES FOR IPLIN
C --READ
CALL READEN(IDETBF,(MNDNDF),ISTAT,
MSALIN,LIMCHI,MSASLO,MSASHI)
IF((ISTAT.NE.'BADF') AND,
((ISTAT.NE.'OFL')} GO TO 430
CALL RDP4T(CBS=CB1,ISTAT,1.4),
"WHILE READING DETECTION FILE")
CALL ERPTAT(10)
"##9##"
"*I/O ERROR IGNORE OUTPUT"
GO TO 500
430
NTLCHR=:
IF((ISTAT.EQ.'BADR') NOTLCHR=?)
C --MASK
CALL MSKPIX(IDETBF, DETBF)

L-347
C --SCREEN & RESAMPLE

400 IF(IDETBF(PHINT).NE.'INT') GO TO 440
    CALL RESPEC(IPRTBF.
    (IPRTBF,IDENTBF,GETINT)
    GO TO 400
440 IF(IDETBF(PHINT).NE.'BYT') GO TO 450
    CALL RESPEC(IPRTBF.
    (IPRTBF,IDENTBF,GETBYT)
    GO TO 400
450 IF(IDETBF(PHINT).NE.'CHR') GO TO 460
    CALL RESPEC(IPRTBF.
    (IPRTBF,IDENTBF,GETICE)
    GO TO 400
460 IF(IDETBF(PHINT).NE.'NIL') GO TO 470
    CALL RESPEC(IPRTBF.
    (IPRTBF,IDENTBF,GETNIL)
    GO TO 400
470 CALL NOPAT1CBOYCS(IPRTBF,PHINT,1,3).
    ' IS INVALID BIN TYPE FOR IDETBF IN MAPRT'1

C --LOOP TO GET NEXT SCAN LINE FOR CURRENT PRINT LINE

480 MSAL: = MSALIN + 1
    GO TO 410

C C SYMBOLIZE PPD CELLS IN CURRENT PRINT BUFFER

500 CALL SYMPRC(IPRTBF)

C C OUTPUT PRINT LINES FROM CURRENT SYMBOLIZED PRINT BUFFER UNTIL SCAN LINE
C FOR NEW PRINT LINE IS AVAILABLE

550 CONTINUE 9 DO UNTIL IPLIN .GE. IPLNKS
    CALL TICK(IPRTBF) 8 INSERT TICKS FOR IPLIN
    CALL CSTY4N(INRPLCS,11,4, IPLN4,5,0')
    NRPLCS=NRPLCS
    HUNIT=HAB(INITLO)
    IPRTBF(IPRECN)-=IPRTBF(IPRECN)+1
    CALL PROFI(HUNIT,
    NRPLCS,4,NRPLCS,4,NRCH,MF,PSPEC,IPRTBF)
    CALL FIXSYM(IPRTBF) 8 REMOVE TICKS FOR IPLIN
    (IPLIN=IPLIN+1)
    IF(IPLIN.GE.IPLMAX) GO TO 940
    GO TO 550
GO TO 980

C C LOOP TO GET NEXT SCAN LINE FOR NEW PRINT LINE

940 (IF(IPLIN.GT.IPLMAX) GO TO 600
    GO TO 400

C C GENERATE AND PRINT BOTTOM SCALE/BORDER FOR UNIT(S) IN CURRENT SECTION

980 CALL SAMSCL(IPRTBF,(IPLMAX),(IPCD),(IPCH))

L-348
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

NUNIT=NTAB(NITLO)
CALL PROVF(NUNIT)

C
C SKIP 2 LINES ON UNIT(S)
C
C GO 700 NIT=NITLO,NITHI
NUNIT=NTAB(NIT)
WRITE(NUNIT,890)
890 FORMAT(0*)
700 CONTINUE
C
C LOOP FOR PRINT COLUMNS IN NEXT SECTION OF CURRENT WINDOW
C
IPCL0-IPCHI+1
IPCHI=MIN(IPCL0+IPCINC-1),IPCMAX
800 CONTINUE
C
C PRINT FCJNOTE ON LAST UNIT OF WINDOW
C
NUNIT=NTAB(NITHI)
WRITE(NUNIT,850)
850 FORMAT('0'/EX,'**SEE UNIT 0 FOR LEGEND**')
C
C INCREMENT THE WINDOW
C
NWND0=NWND0+1
C
C RETURN TO CALLING ROUTINE
C
900 RETURN
C
C
C
C
C
C
C INTERNAL
SUBROUTINE INITIC: & INITIALIZE TICK POINTER AND GET FIRST TICK
O IPLTIC:  & PRINT LINE FOR TICK
O IPCTIC:  & PRINT COLUMN FOR TICK
O JSYTIC:  & TICK SYMBOL: '*'-PRIMARY, '+'-SECONDARY
C
C HISTORY
C
E H SCHLOSSER  LEC  07/22/73  ORIGINAL CODE
E H SCHLOSSER  LEC  03/19/79  'DEN'/'RAD'/'CLA' DETECTION FILES
D A BECK  LEC  12/30/79  REVISE FOR PXBDEF FORMAT BUFFERS
C
C METHOD

L-349
INITIC: INITIALIZE TICK POINTER AND OBTAIN FIRST TICK.
GETIC: OBTAIN NEXT TICK.

MACHINE-DEPENDENT CODE
NONE.

EXTERNAL REFERENCES
NONE.

EXCEPTIONS
1. INITIC MUST BE CALLED BEFORE ANY CALL TO GETIC.

GLOBAL DECLARATIONS
INCLUDE KOMTBL.LIST  & COMMON TICK TABLE AND FUNCTIONS

LOCAL DECLARATIONS
INTEGER JSYD1(/'•', '.'/)'  & TICK SYMBOLS FOR TICK LEVELS 0 & 1
INTEGER NTICK   & TICK POINTER

PROCEDURE

BEGIN 

CALL INITIC; THEN GET FIRST TICK

NTICK=0

ENTRY GETIC; & GET NEXT TICK
0 IPLTIC.  & INTEGER PRINT LINE FOR TICK
0 IPCTIC.  & INTEGER PRINT COLUMN FOR TICK
0 JSYTIC)  & TICK SYMBOL: '.'-PRIMARY, '•'-SECONDARY

INCREMENT TICK POINTER
NTICK=NTICK+1

GET TICK LINE, COLUMN, AND LEVEL
IPLTIC=LINTIC(NTICK)

L-350
IPCLIC=COLTIC(INTICK)
LVLTIC=LEVLTIC(INTICK)

C C TRANSFORM TICK LEVEL TO SYMBOL
C
JSYIC=JSY1C(LVLTIC+1)

C C RETURN TO CALLING ROUTINE
C
900 RETURN

C C INTERNAL
SUBROUTINE SAMSCL: 8 GENERATE SAMPLE SCALE AND BORDER
0 IPRTBF. 8 PRINT BUFFER
1 IPL1N. 8 PRINT LINE
( IPCM1N. 8 MINIMUM PRINT COLUMN
( IPCM1AX) 8 MAXIMUM PRINT COLUMN

C C HISTORY
C
E H SCHLOSSER LEC 07/22/73 ORIGINAL CODE
E H SCHLOSSER LEC 03/19/79 'DEN'/'RAD'/'CLA' DETECTION FILES
D A BECK LEC 12/30/79 REVISE FOR PX8DEF FORMAT BUFFERS

C C METHOD
SET OUTPUT BUFFER PREAMBLE, ENCODE COLUMN NUMBERS AND INSERT COLON, STRING, AND COLON INTO PRINT BUFFER.

C C MACHINE-DEPENDENT CODE
ASSUMES 8 CHARACTERS PER INTEGER BIN.

C C EXTERNAL REFERENCES
PUTC1HR 8 PUT CHAR IN CHAR STRING
CSTN1N 8 CHARACTER STRING FOR INTEGER

C C EXCEPTION
1. ASSUMES 8 CHARACTERS PER INTEGER BIN.

C C GLOBAL DECLARATIONS

L-351
INCLUDE PXBDLF.LIST

DEFINE PIXEL BUFFER STRUCTURE

C LOCAL DECLARATIONS

INTEGER IPRTBF (1) 8 ARGUMENT
INTEGER IPCOL 8 PRINT COLUMN NUMBER
INTEGER IPBIN 8 PRINT BUFFER BIN NUMBER

C PROCEDURE

C SET PREAMBLE POINTERS

IPRTBF (PXLINO) = IPLIN
IPRTBF (PXCHAN) = 0
IPRTBF (PXQUAL) = 0
IPRTBF (PXBININT) = 'INT'
IPRTBF (PXLBIN) = 2
IPRTBF (PXLCOL) = IPCMIN
IPRTBF (PXHBIN) = IPCHMAX - IPCMIN + 2
IPRTBF (PXHCOL) = IPCHMAX
IPRTBF (PXNOIN) = 0
IPRTBF (PXNOAD) = 0
IPRTBF (PXJ01) = 0
IPRTBF (PXHJ01) = 0

C INSERT COLON, ENCODED COLUMN NUMBER, AND COLON INTO BUFFER

IPBIN = IPRTBF (PXLBIN)
DO 100 IPCOL = IPCMIN, IPCHMAX
   CALL PUTCNR (IPRTBF (PXBINS - 1 + IPBIN) (1), ' : ')
   CALL CSTV (IPRTBF (PXBINS - 1 + IPBIN) (2), IPCOL, ' 0 ')
   CALL PUTCNR (IPRTBF (PXBINS - 1 + IPBIN) (6), ' : ')
   IPBIN = IPBIN + 1
100 CONTINUE

C RETURN TO CALLING ROUTINE

900 RETURN

C INTERNAL

SUBROUTINE SYMPRC (IPRTBF) 8 SYMBOLIZE PRINT BUFFER
   U (IPRTBF) 8 PRINT BUFFER

L-352
DAN PACKAOE APPENDIX L
MAIN PROGRAMS/ROUTINES

HISTORY

E N SCHLOSSER LEC 07/22/73 ORIGINAL CODE
E N SCHLOSSER LEC 03/19/79 'OEN''/''RAD''/''CLA' DETECTION FILES
D A BECK LEC 12/30/79 REVISE FOR PXBDEF FORMAT BUFFERS

METHOD

LOOK UP SYMBOLS AND SYMBOLIZE THE PRINT BUFFER.

EXTERNAL REFERENCES

NONE.

EXCEPTIONS

NONE.

GLOBAL DECLARATIONS

INCLUDE PXBDEF.LIST 8 DEFINE PIXEL BUFFER STRUCTURE
INCLUDE KOMSYM.LIST 8 COMMON SYMBOL TABLE

LOCAL DECLARATIONS

INTEGER IPRTBF(1) 8 ARGUMENT
INTEGER IPBIN, IPLBIN, IPHBIN 8 BIN LOW, MIDDLE, AND HIGH

PROCEDURE

SET BIN POINTERS

IPLBIN=IPRTBF(PXLBIN)
IPHBIN=IPRTBF(PXHBIN)

LOOK UP SYMBOLS FOR SYMBOLIZATION OF BUFFER

DO 150 IPBIN=IPLBIN, IPHBIN
   IPXEL=KSYM(IPRTBF(PXBINS-1+IPBIN)+KSYMSZ-1)
   IPRTBF(PXBINS-1+IPBIN)=KSYM(IPXEL+1)
150 CONTINUE

SET PRINT SPEC TO ALLOW OVERPRINTING

JPSPEC='1000***'

L-353
INTERNAL
SUBROUTINE PRSINII  / INITIALIZE PRINT BUFFER
U  IPRTBF.  / PRINT BUFFER
I  IPLIN.  / PRINT LINE
I  IPCMIN.  / MINIMUM PRINT COLUMN
I  IPCMAX.  / MAXIMUM PRINT COLUMN

HISTORY
E W SCHLOSSER  LEC  07/22/73  ORIGINAL CODE
E W SCHLOSSER  LEC  03/19/79  'DEN'/ 'RAD'/ 'CLA' DETECTION FILES
D A BECK  LEC  12/30/79  REVISE FOR PXBDEF FORMAT BUFFERS

METHOD
SET 'NO INFO' FLAG & 'NO DATA' THRESHOLD VALUES DEPENDING ON RESAMPLING
ALGORITHM. SET THE PRINT BUFFER PREAMBLE AND INITIALIZE THE BINS
TO 'NO INFO'.

MACHINE-DEPENDENT CODE
NONE.

EXTERNAL REFERENCES
NONE.

EXCEPTIONS
NONE.

GLOBAL DECLARATIONS
INCLUDE PXBDEF.LIST  / DEFINE PIXEL BUFFER STRUCTURE
INCLUDE KOMSYM.LIST  / COMMON SYMBOL TABLE

LOCAL DECLARATIONS
INTEGER IPBIN, IPLBIN, IPHBIN  / BIN #, LOW, AND HIGH
INTEGER NODAPP, NOINPP  & 'NO DATA', 'NO INFO'
INTEGER IPRTBF(1)  & ARGUMENT

C
C PROCEDURE
C
C SET 'NO DATA'/'NO INFO' VALUES DEPENDING ON TYPE OF RESAMPLING
C
NODAPP=-9999
NOINPP=0
IF(KT1PIX.EQ.0) NOINPP=ISYMHI+1

C
C SET PREAMBLE POINTERS
C
IPRTBF(PXLIN0)=IPLIN
IPRTBF(PXCHAN)=0
IPRTBF(PXQUAL)=0
IPRTBF(PXBINI)= 'INT'
IPRTBF(PXLBIN)=2  & TICK HALO ALGORITHM CHECKS TICK BIN -1
IPRTBF(PXLIN(COL))=IPC(MIN
IPRTBF(PXBIN(COL))=IPCMAX-IPCMIN+2
IPRTBF(PXH(COL))=IPCMAX
IPRTBF(PXNOIN)=NOINPP
IPRTBF(PXNOI)=NODAPP
IPRTBF(PXLJOI)=0
IPRTBF(PXNOJ)=0

C
C INITIALIZE BINS TO 'NO INFO' VALUE
C
IPBIN=IPRTBF(PXLBIN)
IPBIN=IPRTBF(PXBIN)
DO 200 IPBIN=IPBIN,IPBIN+1
IPRTBF(PXBINS(IPBIN)+NOINPP
200 CONTINUE

C
C RETURN TO CALLING ROUTINE
C
900 RETURN

C
C
C INTERNAL
SUBROUTINE TICK  & INSERT TICKS
U (IPRTBF)  & PRINT BUFFER
C
C HISTORY
C
E H SCHLOSSER  LEC  07/22/73  ORIGINAL CODE

L-359
EXTERNAL REFERENCES

INTEGER ICE 8 INTEGER-CHARACTER-EQUIVALENT OF 1ST CHAR IN STRING

EXCEPTIONS

1. INITIC MUST BE CALLED BEFORE FIRST CALL TO TICK ROUTINE.

LOCAL DECLARATIONS

INTEGER IPRTBF(1) 8 ARGUMENT
INTEGER IPBTIC 8 BIN # OF TICK
INTEGER ICESP A 8 I-C-E OF ''
INTEGER NBINCO 8 BIN # OF COLUMN ZERO

PROCEDURE

INITIALIZE POINT FOR SAVE TABLE OF SYMBOLS REPLACED BY TICKS

NBINCO=IPRTBF(PXLBIN)-IPRTBF(PXLCOL) 8 FOR INSERTING TICKS
NSAVED=0
ICESP A=ICE('')

GET AND INSERT TICKS

500 IPBTIC=IPCTIC+NBINCO
   IF (IPBTIC.GT.IPLIN) GO TO 900 8 SAVE TICK FOR SUBSEQUENT LINE
   IF (1
      & (IPBTIC.LT.IPLIN).OR.
      & (IPBTIC.LT.IPRTBF(PXLBIN)).OR.
      & (IPBTIC.GT.IPRTBF(PXHBNIN)))
         8 TICK ABOVE PRINT LINE
         8 TICK LEFT OF LOW BIN
         8 TICK RIGHT OF HIGH BIN
8 GO TO 590
   0 THEN IGNORE IT
   IF (JSY Tic.EQ. '*') GO TO 510  0 ALWAYS INSERT PRIMARY TICK
   IF (ICE(IPRTBF(PXIBINS-1+IPBTIC).NE.1.EQ.1)) GO TO 510 0 PUT SEC TICK
   IF (IPRTBF(PXIBINS-1+IPBTIC).NE.1.EQ.1) GO TO 590 0 PUT SEC TICK IN NO DAT
   CONTINUE  0 INSERT LEFT HALO IN 'NO DATA' PPD CELL
   IF (IPRTBF(PXIBINS-1+IPBTIC).NE.1.EQ.1) GO TO 520
      NSAVE=NSAVD(NSAVED+1,NSAVMX)
      ISAVSY(NSAVED.1)=IPBTIC-1
      ISAVSY(NSAVED.2)=IPRTBF(PXIBINS-1+IPBTIC-1)
      IPRTBF(PXIBINS-1+IPBTIC-1)=''
   520 CONTINUE  0 INSERT TICK IN PPD CELL
      NSAVE=NSAVD(NSAVED+1,NSAVMX)
      ISAVSY(NSAVED.1)=IPBTIC
      ISAVSY(NSAVED.2)=IPRTBF(PXIBINS-1+IPBTIC)
      IPRTBF(PXIBINS-1+IPBTIC)=JSY Tic
   530 CONTINUE  0 INSERT RIGHT HALO IN 'NO DATA' PPD CELL
   IF (IPRTBF(PXIBINS-1+IPBTIC).NE.1.EQ.1) GO TO 590
      NSAVE=NSAVD(NSAVED+1,NSAVMX)
      ISAVSY(NSAVED.1)=IPBTIC+1
      ISAVSY(NSAVED.2)=IPRTBF(PXIBINS-1+IPBTIC+1)
      IPRTBF(PXIBINS-1+IPBTIC+1)=''
      9 RIGHT HALO
   590 CALL GETIC (IPRTIC,IPB1C,JSY Tic)
   GO TO 590
C
C RETURN TO CALLING ROUTINE
C
C 900 RETURN
C
C C C
C INTERNAL
C SUBROUTINE FIXSY M ( 8 UNTICK AND FIX SYMBOLS IN PPD BUFFER
U (IPRTBF) 8 PRINT BUFFER
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
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
C REPLACE TICK(S) WITH ORIGINAL SYMBOL(S).
C
C IF COUNTING, SET NON-BLANK CELLS TO ':' (NO DATA) AND SET PRINT
C
SPEC TO SUPPRESS OVERPRINTING.

MACHINE-DEPENDENT CODE

ASSUMES 6 CHARS PER INTEGER BIN.

EXTERNAL REFERENCES

INTEGER ICE  8 INTEGER-CHARACTER-EQUIVALENT OF 1ST CHAR IN STRING

EXCEPTIONS

NONE.

GLOBAL DECLARATIONS

INCLUDE PXBDEF.LIST  8 DEFINE PIXEL BUFFER STRUCTURE
INCLUDE KOMKLIS.LIST 8 COMMON CLASSIFICATION INFO

LOCAL DECLARATIONS

INTEGER IPRT0F(1)  8 ARGUMENT
INTEGER ICESP A  8 1-C-E OF • •
INTEGER IPBIN. IPLBIN. IPHBIN  8 BIN •. LOW. AND HIGH

PROCEDURE

REPLACE TICK(S) & TICK HALO(S), IF ANY, WITH ORIGINAL SYMBOL(S)

200 IF(INSAVED.LE.0) GO TO 500
   IPBIN=ISAVSY(INSAVED,1)
   IPRT0F(PXBSNS-1+IPBIN)-ISAVSY(INSAVED,2)
   NSAVID=NSAVED-1
   GO TO 200

500 IF(KTIPX.EQ.0) GO TO 900
   IPLBIN=IPRT0F(PXLSBN)
   IPHBN=IPRT0F(PXBSBN)
   ICESP A-ICE('••••')
   GO 700 IPBIN=IPLBIN. IPHBN
   IF(ICE(IPRT0F(PXBSNS-1+IPBIN))1).NE. ICESP A)
      IPRT0F(PXBSNS-1+IPBIN)-'••••'
    700 CONTINUE
      JPSCP='••••••'  8 NO NEED TO EVEN ATTEMPT OVERPRINTING

L-398
C RETURN TO CALLING ROUTINE
C 900 RETURN
END
SUBROUTINE NITNOO 8 PRINT MAP UNIT HEADING
I HUNIT) 8 OUTPUT PRINT UNIT

HISTORY

E H SCHLOSSER LEC 07/20/73 ORIGINAL CODE
E H SCHLOSSER LEMSCO 01/20/80 CALL PRNUM INSTEAD OF PRNUM
E H SCHLOSSER LEMSCO 09/18/80 NEGATIVE MAP NUMBER & USE GINET

METHOD

WRITE BOX HEADING, SCENE IDENTIFICATION, CLASSIFICATION SUMMARY,
MAP WINDOW IDENTIFICATION, AND CONTROL SUMMARY ON HUNIT.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

HUNIT 8 PAGE EJECT & PRINT PAGE HEADING LINES ON SPECIFIED UNIT
GETICE 8 GET INTEGER-CHARACTER-EQUIVALENT FROM CHAR IN STRING
PRNUM 8 PRINT BOX NUMBERS ON SPECIFIED UNIT
CLSCNO 8 PRINT CLASSIFICATION HEADING ON SPECIFIED UNIT
MAPPNO 8 PRINT MAP WINDOW HEADING INFORMATION ON SPECIFIED UNIT

EXCEPTIONS

NONE.

GLOBAL DECLARATIONS

INCLUDE KOMNULIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMALT.LIST 8 COMMON ALTERNATE PRINT FILE POINTERS

LOCAL DECLARATIONS

INTEGER IMI 8 HOURS UNITS DIGIT
INTEGER IHI 8 MINUTES TENS DIGIT
INTEGER IHI 8 MINUTES UNITS DIGIT

L-350
INTEGER WND100 0 WINDOW NUMBER HUNDREDS DIGIT
INTEGER WND10 0 WINDOW NUMBER TENS DIGIT
INTEGER WND1 0 WINDOW NUMBER UNITS DIGIT
INTEGER WNDICE 0 INTEGER-CHARACTER-EQUIVALENT OF '0'
PARAMETER BLANK=99 0 BOX NUMBER PRINTS AS BLANK

PROCEDURE
-------

CALL TRACE

BEGIN HEADING

PAGE=0
CALL NUNIT(4,NUNIT)

SET UP HOUR AND MINUTE BOX NUMBERS

ICED=ICED('0')
CALL GETICE(INH, JHMS.2)
INH=INH-ICED
CALL GETICE(INH, JHMS.3)
INH=INH-ICED
CALL GETICE(INH, JHMS.4)
INH=INH-ICED

SET UP MAP NUMBER BOX NUMBERS

WIND00=WIND00/10 0 100'S DIGIT
WIND10=WIND00*100*WND100/10 0 10'S DIGIT
WND1=WIND00-10*WND10 0 1'S DIGIT
WND100=WIND100+10 0 NEGATIVE (WRITE ON BLACK)
WND10=WIND10+10
WND1=WIND1+10

PUT BOX NUMBERS ON WIDE PAGE

IF (PAGE.LT.120) GO TO 300
CALL PNUMUNIT(16,12,INH,INH,WND100,WND10,WND1,NIT,BLANK)
WRITE(NUNIT,135)

135 FORMAT(1X,
1 8X,'HOUR'.13X,'MINUTE'.2X,'MAP NUMBER'.2X,'UNIT')
GO TO 700

PUT BOX NUMBERS ON NARROW PAGE

300 IF (PAGE.LT.80) GO TO 500
CALL PNUMUNIT(16,12,WND100,WND10,WND1,NIT,BLANK,BLANK,BLANK)

L-301
WRITE (NUNIT, 339)
339 FORMAT (1X/
  1  8X,49(" ").10X,0(" ")/
  2  20X,\"MAP NUMBER\",20X,\"UNIT\")
GO TO 700

C
C PUT BOX NUMBERS ON VERY NARROW PAGE
C
500 CALL PRNUM (NUNIT, 18, 18, 14, 14, BLANK, BLANK, BLANK, BLANK)
WRITE (NUNIT, 339)
339 FORMAT (1X/
  1  8X,26(" ").10X,0(" ")/
  2  16X,\"MAP NUMBER\",20X,\"UNIT\")

C
C COMPLETE HEADING
C
700 CALL CLSNO (NUNIT)
CALL MAPHDG (NUNIT)
NLINE=99
RETURN
END
SUBROUTINE OPRPRC  & OPEN ALTERNATE PRINT FILES FOR PRTCLASS

HISTORY

E M SCHLOSSER  LEC  08/29/78  ORIGINAL CODE
J C CRISP  LEC  12/18/79  REVISE TO INCLUDE PRCDEF-PROC

METHOD

A MAXIMUM OF 9 ALTERNATE PRINT FILES ARE OPENED & INITIALIZED.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

OPENPR  OPEN ALTERNATE PRINT FILES

EXCEPTIONS

1. ONE PRINT FILE IS ALWAYS OPENED, UNLESS IN DATA/CHECKOUT MODE.

GLOBAL DECLARATIONS

INCLUDE KOMXOT.LIST  & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE PRCDEF.LIST & DEFINE MALTHI AND KPAGHI FOR PRTCLASS

PROCEDURE

CALL TRACE

NO MORE THAN MALTHI ALTERNATE PRINT FILES FOR PRTCLASS
MALTHI-MAX(MALTHI,I)
MALTHI-MIN(MALTHI,MALTHI)

NO MORE THAN KPAGHI COLUMNS PER PAGE IN PRTCLASS
DOH PACKAEE APPENDIX L
MAIN PROGRAMS/ROUTINES

KPAGC=MAX0(KPAGC,1)
KPAGC=MIN0(KPAGC,KPAGC)

C OPEN FILE(S)
C IF(MDATAC.EQ.0) CALL OPENPR
C
RETURN
END
SUBROUTINE RESPRC: SCREEN/RESAMPLE/COUNT/STORE GET PIXELS FOR PRCLASS
O IPRTBF. 0 UNPACKED RESAMPLED PPD CELLS
I IPBUF. 0 PRINT BUFFER
I IDETBF. 0 PACKED UNRESAMPLED DETECTION PIXELS
I GETBIN) 0 ROUTINE TO GET PIXEL VALUE (OETBYT, OETCHR, OETINT)

HISTORY
-----

CPU

E H SCHLOSSER LEC 07/02/73 ORIGINAL CODE IN MAPRNT
E H SCHLOSSER LEC 03/15/79 REVISE & SEPARATE INTO REOPRC
J C CRISP LEC 12/12/79 REVISE/RENAME FOR PXBDEF BUFFER

METHOD
-----

COMPUTE BIN NUMBERS FOR SAMPLE AND COLUMN 0.
COMPUTE SAMPLES FOR LOW AND HIGH PRINT COLUMNS. IF LOW SAMPLE
GREATER THAN LAST DEFINED SAMPLE, INSERT PPD NODATA VALUE IN BUFFER
PPD CELLS AND RETURN. IF LOW SAMPLE IS LESS THAN FIRST DEFINED SAMPLE
INSERT PPD NODATA VALUE IN PPD CELLS UNTIL FIRST DEFINED SAMPLE IS
FOUND. GET SAMPLE VALUE AND CHECK AGAINST DETECTION NODATA THRESHOLD.
RESAMPLE ALL INPUT NODATA PIXELS AND STORE THE PPD NODATA THRESHOLD
IN THEIR PPD CELLS.

SCREEN ALL INPUT DATA PIXELS AGAINST LCVLO(11) & LCVHI(11). IGNORE PIXELS
OUTSIDE THESE LIMITS, RESAMPLE EACH SCREENED DATA PIXEL AND
IF KTIPIX IS NON-ZERO:
ADD VALUE OF THE DATA PIXEL (BUT NOT MORE THAN KTIPIX) TO THE
CORRESPONDING OUTPUT PPD CELL.
OTHERWISE:
STORE VALUE OF THE DATA PIXEL IN THE PPD CELL.

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

UTILIZES UNIVAC FORTRAN V FUNCTION 'LOC'.

EXTERNAL REFERENCES
---------------------

OETBYT 0 GET VALUE OF BYTE FROM BYTE STRING
OETCHR 0 GET INTEGER-CHARACTER-EQUIVALENT FROM CHAR STRING
OETINT 0 GET VALUE OF INTEGER FROM INTEGER STRING

EXCEPTIONS
----------

1. IF THE SAME ACTUAL ARGUMENTS ARE NOT USED FOR IPRTBF AND IPBUF.
A FATAL ERROR WILL BE ISSUED.

GLOBAL DECLARATIONS

INCLUDE CONNER.LIST  COMMON SCENE PARAMETERS
INCLUDE KOMFIT.LIST  COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMKLTS.LIST  COMMON CLASSIFICATION INFO
INCLUDE TRFORM.LIST  DEFINE COORDINATE TRANSFORMATION FUNCTIONS
INCLUDE PXBDEF.LIST  DEFINE PIXEL BUFFER STRUCTURE

LOCAL DECLARATIONS

INTEGER IPRTBF(1)  ARGUMENT
INTEGER IDETBF(1)  ARGUMENT
INTEGER MSASAM  SAMPLE NUMBER OF PIXEL BEING SCREENED/RESAMPLED
INTEGER MSALIN  MSS LINE NUMBER
INTEGER MSASLO  MSS LOW SAMPLE
INTEGER MSASHI  MSS HIGH SAMPLE
INTEGER NOODAE  DETECTION BUFFER NO DATA THRESHOLD
INTEGER NOPR  PRINT BUFFER NO DATA THRESHOLD
INTEGER NRC  RADIANCE/DENSITY/CLASS BEING SCREENED/RESAMPLED
REAL CORLIN  CORRECTED LINE NUMBER FOR MSALIN
REAL CORSAM  CORRECTED SAMPLE NUMBER FOR MSASAM
REAL ADLIN  ADJUSTED LINE NUMBER FOR MSALIN
REAL ADJSAM  ADJUSTED SAMPLE NUMBER FOR MSASAM
INTEGER IPCL0.IPCHI  LOW AND HIGH PRINT COLUMNS
INTEGER IPLIN  PRINT LINE
INTEGER IPCOL  PPY COLUMN FOR MSASAM
INTEGER IPCL.IPC2  PRINT COLUMNS FOR FIRST AND LAST NO DATA PIXELS
INTEGER NBINSO  BIN CONTAINING SAMPLE 0 IN IDETBF
INTEGER NBINCO  BIN CONTAINING COLUMN 0 IN IPRTBF
INTEGER IPBIN  BIN POINTER IN IPRTBF

PROCEDURE

CHECK THAT SAME ACTUAL ARG WAS USED FOR IPRTBF AND IPBUF

IF(LOC(IPRTBF),NE.,LOC(IPBUF)) CALL MDFAIL(
 = "IPRTBF & IPBUF NOT SAME IN RESPRC")

COMPUTE BIN NUMBER FOR SAMPLE AND COLUMN 0 AND BIN POINTER

NBINSO=IDETBF(PXLBIN)-IDENTBF(PXLSAM)
NBINCO=IPRTBF(PXLBIN)-IPRTBF(PXLCOL)
IPBIN=PXBINS-1+NBINCO

L-388
C
C 150 CALL GETBIN ( NRDC, IDETBF ( PXBINS ) , ( MSASAM + NBINS ) )
C 150 IF ( NRDC . GE . NODE ) GO TO 500
C
C
C
L-367
DATA STORE SUB-STATE (SCREEN/RESAMPLE/STORE DATA PIXELS)

300 IF((NRDC.LT.LCVLO(1)).OR.(NRDC.GT.LCVHI(1))) GO TO 320
   IMRTBF(IPBIN+IPCOL)=NRDC & IN RAD/DEG/CLA LIMITS
320 MSASAM=MSASAM+1
   CORSAM=CORSAM(NSALIN,MSASAM)
   IPCOL=IPCOL(CORLIN,CORSAM)
   IF(MSASAM.GT.MSASHI) GO TO 900
   IF(IPCOL.GT.IPCHI) GO TO 900
   CALL GETBIN(NRDC, IPBF,PXSBINS1,MSASHAM,MSAMNSO)
   IF(NRDC.GE.NODADE) GO TO 300
   GO TO 500

DATA COUNT SUB-STATE (SCREEN/RESAMPLE/COUNT DATA PIXELS)

400 IF((NRDC.LT.LCVLO(1)).OR.(NRDC.GT.LCVHI(1))) GO TO 420
   IMRTBF(IPBIN+IPCOL)=N
   IMRTBF(IPBIN+IPCOL)+MINTX(NRDC.KTIPIX) & IN RAD/DEG/CLA LIMITS
420 MSASAM=MSASAM+1
   CORSAM=CORSAM(NSALIN,MSASAM)
   IPCOL=IPCOL(CORLIN,CORSAM)
   IF(MSASAM.GT.MSASHI) GO TO 700
   IF(IPCOL.GT.IPCHI) GO TO 900
   CALL GETBIN(NRDC, IPBF,PXSBINS1,MSASHAM,MSAMNSO)
   IF(NRDC.GE.NODADE) GO TO 400

NODATA STATE (RESAMPLE/STORE DEFINED NODATA PIXELS)

500 IMRTBF(IPBIN+IPCOL)=NODAPR
   MSASAM=MSASAM+1
   CORSAM=CORSAM(NSALIN,MSASAM)
   IPCOL=IPCOL(CORLIN,CORSAM)
   IF(MSASAM.GT.MSASHI) GO TO 700
   IF(IPCOL.GT.IPCHI) GO TO 900
   CALL GETBIN(NRDC, IPBF,PXSBINS1,MSASHAM,MSAMNSO)
   IF(NRDC.GE.NODADE) GO TO 500
   GO TO 200

FINAL UNDEFINED STATE (RESAMPLE/STORE FINAL UNDEFINED PIXELS)

700 IPC1=IPCOL
   CORSAM=CORSAM(NSALIN,MSASHI)
   IPC2=IPCOL(CORLIN,CORSAM)
   DO 750 IPCOL=IPCL,IPC2
   IF(IPCOL.GT.IPCHI) GO TO 900
   IMRTBF(IPBIN+IPCOL)=NODAPR
750 CONTINUE
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

C
C FINAL STATE (ALL DONE)
C 900 RETURN
END
PROGRAM PLTCLASS

(R E NARVESON)

This program produces plots of classified ERTS MSS data on a pen plotter. The user specifies the scale, window dimensions, location, etc. of each map.

This program is limited to that part of the world covered by the Clarke 1866 spheroid (North America).

The program CLASSIFY must be executed before this program, preferably in the same run.

```
INCLUDE KOMXQT.LIST
INCLUDE KOMLOG
INCLUDE KOMLUS
INCLUDE KOMLEN
INCLUDE KOMNER
INCLUDE KOMLKS
INCLUDE KOMFIT
INCLUDE KOMDET
INCLUDE KOMALT
INCLUDE KOMSYM
INCLUDE KOMPLT
INCLUDE KOMIH
INCLUDE KOMOM

EXTERNAL PLC000.PLLXQT

CALL NVIATO( PLC000.PLLXQT )  // FIRST CALL IS VIA PLC000 TO PLLXQT
100 CALL VIATO
     GO TO 100
END
```
PROGRAM PLTCLASS/VIRTUAL

HISTORY

E N SCHLOSSER LEC 08/08/74 ORIGINAL CODE
E N SCHLOSSER LEC 11/08/78 SNAP.FZ(INI: NO 'N' IN DEMAND)

METHOD

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

MACHINE-DEPENDENT CODE

WRITTEN IN ASSEMBLER FOR THE UNIVAC 1100 SERIES COMPUTERS UNDER THE EXEC-9 OPERATING SYSTEM USING 6-BIT FIELD DATA 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-9 CONTROL STATEMENT
ER IONS 8 INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS 8 TERMINATE PROGRAM EXECUTION
OAM.PLTCASS-MAP 8 SYMBOLIC MAP DIRECTIVES TO LINK EDIT REAL ABSOLUTE
OAM.SYS-MAPOPT 8 STANDARD MAP OPTIONS WHEN LINK EDITING

EXCEPTIONS

1. RESULTS ARE UNDEFINED UNLESS THE FILE DAM. IS 8ASO-D & 8PREP-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

EXOT OPTIONS ARE PRE-LOADED BY EXEC INTO REGISTER AS IN
**MASTER BIT NOTATION.**

**LOCAL DECLARATIONS**

---

AXRS

$00$. D-BANK

SSSH FORM 6.6.18

LABSDF $00$. 09.1. 'F'. 0. LABEL. I WD. FORTRAN. FILEDATA

LABIMD 'SOFF'.

MAPSDF $00$. 09.0.0. DATA. 9 WDS. FIELDATA

MAPIMO '8XQTS: MAP. FZM. DAM. PLTCLASS-MAP. PLTCLASS'. '8XQTS'

ADDSDF $00$. 09.0.0.

ADDIMO '8XQTS: ADD'. DAM. SYS-MAPOPT

XQTSDF $00$. 09.0.0.

XQTIMO '8XQTS: XQT_1'. PLTCLASS

EOFSDF - 0. END-OF-FILE STOP WORD

PF FORM 12.6.18

CSFASO 'SASOT 20'.

CSFAOD 'SADD 20'.

SAVREG RES 1

1OPKT 1500 '20'. WS 33. LABSDF. '0' 0

---

**PROCEDURE**

---

$01$. D-BANK

PLTCLASS* LA U

A0.4. A0. + . SKIP NEXT INST IF A0.<4 (NOT DEMAND)

THE. U

A0.4. A0. MAPIMO+2. DEMAND: BLANK OUT N OPTION

SA. 32. A0. (CSFASO)

LA A0. (CSFASO)

ER CSFS

A0. SAVREG.

STORE &

PSRINT (PF 2.1. SAVREG). PRINT SASO STATUS

GETOPT

LOAD OPT LTRS INTO A2. A3. A4

PUTOPT 05

A2. XQTSDF+2. STORE OPTION LETTERS INTO XQTS IMAGE

SA A4. XQTSDF+4. (13 WORDS -- MAX 10 OPT LETTERS)

WRITE LA A0. (1OPKT)

ER 30WS. WRITE SDF IMAGES TO 20.

ADD LA A0. (CSFASO)

ER CSFS

ER EXITS

END PLTCLASS
PLTCCLASS-Map

PLTCCLASS OVERLAY STRUCTURE

HISTORY

R E Naryeson LEC 11/07/79 MODIFIED FROM PRTCLASS-MAP
E H Schlosser LEC 02/20/79 MACRO COMMANDS & TIME COMMAND

LIB DAM.

SEO S-MAIN
IN DAM.PLTCCLASS/ MAIN PROGRAM
IN DAM.NVIAO MAIN CALL 'VIA' AND 'TO' SUBROUTINES
IN DAM.NULSUB DO NOTHING
IN DAM.SYS-BLOCK BLOCK DATA SUBROUTINE

IN OAM.PL0000 CALL USER-SPECIFIED PHASE 0 ROUTINE
IN OAM.PLCIBS CALL PREVIOUSLY NAMED PHASE 1/2/9 'TO' ROUTINE
IN OAM.NTABS/DAM DAM UNIT & TABLE GOES IN SAME SEO W/ FORTRAN I/O

SEO START-STOP
IN DAM.PLXQT PLTCCLASS INITIALIZATION ROUTINE
IN DAM.PLCEXI PLTCCLASS TERMINATION ROUTINE

SEO SPECIFY*, START-STOP
IN DAM.KXDCLE CLEAR WARNINGS/ERRORS
IN DAM.KXDCOP GET/CHECK NUMBER OF OUTPUT COPIES
IN DAM.KXDDEN GET/CHECK DENSITY LIMITS
IN DAM.KXDEXP EXPLAIN PROGRAM/COMMAND
IN DAM.KXDEHA GET/CHECK PAGE HEADER(S)
IN DAM.KXDER Get/CHECK TRANSVERSE MERCATOR CENTRAL MERIDIAN
IN DAM.KXDEH Get/CHECK WINDOW ENVELOPE/VERTICES
IN DAM.KXDOF TURN OFF MODE SWITCHES
IN DAM.KXDOM TURN ON MODE SWITCHES
IN DAM.KXDOX CONDITIONALLY PERFORM NEXT COMMAND
IN DAM.KXDOF GET/Check WINDOW ORIGIN
IN DAM.KXDOA SKIP TO TOP OF NEXT PAGE
IN DAM.KXDPAS GET/Check PLOTTED SPECIFICATIONS
IN DAM.KXDPAS RENUMBER GET/Check NEW WINDOW SEQUENCE NUMBER
IN DAM.KXDSVE GET/Check WINDOW Envelope/VERTICES
IN DAM.KXDSVE GET/Check SYMBOLS
IN DAM.KXDSVA GET/Check WINDOW SCALE
IN DAM.KXSYN GET/Check WINDOW SCALE
IN DAM.KXSYN GET/Check SYMBOLS
IN DAM.KXSYT GET/Check TICK UNITS/INTERVALS
IN DAM.KXSIW Print CLOCK TIME & CHARGE TIME
IN DAM.KXMIN GET/Check WINDOW ENVELOPE/VERTICES
IN DAM.KXDNX MACRO COMMANDS
IN DAM.KXICON GET/Check UTH PROJECTION ZONE

SEO MAPOUT*, START-STOP
IN DAM.PLCHAP MAP RADIANCE/DENSITY/CLASS (PHASE 0)

L-373
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

IN DAM.PLTCASS/VIRTUAL
SUBROUTINE PLC0001  8 CALL PHASE 8 SUBROUTINES FOR PLCLASS
 1 NARSUB)  8 NAME OF SUBROUTINE TO CALL (OR NULLSUB)

HISTORY

R E HARVEISON  LEC  11/07/78  MODIFIED FROM PLC000
E H SCHLOSSER  LEC  02/20/78  MACRO COMMANDS & TIME COMMAND

METHOD

NEXT COMMAND IS RETRIEVED, VALIDATED, AND ITS SUBROUTINE CALLED.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

READS  8 READ PUNCHED CARD OR TERMINAL INPUT INTO BUFFER
GETBL  8 GET ALPHABETIC COMMAND FROM BUFFER
INTEGER ICE  8 INTEGER-CM-CHAR-EQUIV FOR CHARACTER
WARNS  8 PRINT/LOG WARNING MESSAGE
PLC...  8 DEDICATED SUBROUTINE FOR COMMAND ... (SEE BELOW)
KMD...  8 COMMON SUBROUTINE FOR COMMAND ... (SEE BELOW)

EXCEPTIONS

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

GLOBAL DECLARATIONS

INCLUDE NULLSTLIST  8 DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

INTEGER KMD  8 FIRST 3 CHAR OF USER COMMAND (BLANK AFTER DONE
INTEGER LSSTAT  8 READS STATUS ('EOF' MEANS END-OF-FILE)
INTEGER KASE  

8 MODIFIED 1-C-E OF FIRST CHAR OF COMMAND

PROCEDURE

----------

CALL PREVIOUSLY NAMED SUBROUTINE

CALL TRACE
CALL NAMSUB  8 CALL TO NULSUB DOES NOTHING

READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)

KOND=' NUL'  8 IMPOSSIBLE INPUT (NOT LEFT JUSTIFIED)
CALL READS(LSSTAT, ' ')  8 FILL BUFFER, BLANK CUE MESSAGE
IF(LSSTAT.NE.'1') KOND='EOF'
IF(KOND.NE.'EOF') CALL OCTSAL(KOND, (3), NULCST)  8 SET 3 ALPHA CHAR

CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT

KASE=IN('KOND')-IN('A')+1  8 A TO Z = 1 TO 26

CASE STATEMENT ON MODIFIED 1-C-E OF COMMAND'S FIRST CHARACTER

IF((KASE.LT.11).OR.(KASE.GT.28)) KASE=27  8 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

DETERMINE COMMAND, PERFORM COMMAND CHANGE KOND TO BLANK

401 CONTINUE 8+++ A
402 CONTINUE 8+++ B
GO TO 800

403 CONTINUE 8+++ C
IF(KOND.EQ.'CLE') CALL KMOCLE(KOND)  8 CLEAR
IF(KOND.EQ.'COP') CALL KMOCOP(KOND)  8 COPIES
GO TO 800

404 CONTINUE 8+++ D
IF(KOND.EQ.'DEN') CALL KMODEN(KOND)  8 DENSITY
GO TO 800

405 CONTINUE 8+++ E
IF(KOND.EQ.'EOF') CALL PLCEX(KOND)  8 END-OF-FILE CAUSES EXIT
IF(KOND.EQ.'EXIT') CALL PLCEX(KOND)  8 EXIT
IF(KOND.EQ.'EXP') CALL KMODEXP(KOND)  8 EXPLAIN
GO TO 800

L-376
C 406 CONTINUE J 405 F
407 CONTINUE 4 00 TO 800
C 408 CONTINUE 8**** M
IF(KOMD.EQ.'HEA') CALL KMDHEA(KOMD) a HEADING
GO TO 800
C 409 CONTINUE 8**** I
410 CONTINUE 8**** J
411 CONTINUE 8**** K
412 CONTINUE 8**** L
GO TO 800
C 413 CONTINUE 8**** M
IF(KOMD.EQ.'MAP') CALL PLCMAP(KOMD) a MAP
IF(KOMD.EQ.'MER') CALL KMDMER(KOMD) a MERIDIAN
GO TO 800
C 414 CONTINUE 8**** H
IF(KOMD.EQ.'NEW') CALL KMDNEW(KOMD) a NEWS
IF(KOMD.EQ.'NEX') CALL KMDNEX(KOMD) a NEXT
GO TO 800
C 415 CONTINUE 8**** O
IF(KOMD.EQ.'OFF') CALL KMDOFF(KOMD) a OFF
IF(KOMD.EQ.'ON') CALL KMDON(KOMD) a ON
IF(KOMD.EQ.'ORI') CALL KMDORI(KOMD) a ORIGIN
GO TO 800
C 418 CONTINUE 8**** P
IF(KOMD.EQ.'PAO') CALL KMDPAG(KOMD) a PAGE
IF(KOMD.EQ.'PLO') CALL KMDPLO(KOMD) a PLOTTER
GO TO 800
C 417 CONTINUE 8**** Q
GO TO 800
C 419 CONTINUE 8**** R
IF(KOMD.EQ.'RAD') CALL KMDRAD(KOMD) a RADIANCE
IF(KOMD.EQ.'REN') CALL KMDREN(KOMD) a RENUMBER
GO TO 800
C 419 CONTINUE 8**** S
IF(KOMD.EQ.'SCA') CALL KMDSCA(KOMD) a SCALE
IF(KOMD.EQ.'SYM') CALL KMDSYM(KOMD) a SYMBOLS
GO TO 800
C 420 CONTINUE 8**** T
IF(KOMD.EQ.'TIC') CALL KMDTIC(KOMD) a TICKS
IF(KOMD.EQ.'TIM') CALL KMDTIM(KOMD) a TIME
GO TO 800
C 421 CONTINUE 8**** U
422 CONTINUE 8**** V
GO TO 800
C 423 CONTINUE **** W
  IF(KOMO.EQ. 'WIN') CALL KMDWIN(KOMO) 8 WINW
GO TO 800
C 424 CONTINUE **** X
C 425 CONTINUE **** Y
GO TO 800
C 426 CONTINUE **** Z
  IF(KOMO.EQ. 'ZON') CALL KMDZON(KOMO) 8 ZONE
GO TO 800
C 427 CONTINUE **** NOT ALPHABETIC
  IF(KOMO.EQ. 'BAD') CALL KMDBAD(KOMO) 8 BAD
  IF(KOMO.EQ. 'SAD') CALL KMDSAD(KOMO) 8 SADD
  IF(KOMO.EQ. 'SAS') CALL KMDSAS(KOMO) 8 SADD
  IF(KOMO.EQ. 'SBR') CALL KMDSBR(KOMO) 8 SBRKPT
  IF(KOMO.EQ. 'SFR') CALL KMDSFR(KOMO) 8 SFREE
  IF(KOMO.EQ. 'SLO') CALL KMDSLO(KOMO) 8 SLO
C  IF COMMAND WAS NOT FOUND, TRY MACRO-COMMAND
C 800 IF(KOMO.NE. 'I') KOMD='PLC-'
  & 8 CHAR NAME PLUS PL0000 NAME
  IF(KOMO.NE. 'I') CALL KMDXXX(KOMO)
C  COMMAND IS INVALID IF STILL NOT FOUND
C  IF(KOMO.NE. 'I') CALL WARNS('INVALID COMMAND --')
C  FORCED ALL FORTRAN I/O ROUTINES INTO SAME SEG AS PL0000 (NEVER PERFORMED)
C  IF(KOMO.EQ. 'JUNK') READ(895,895) KOMO
  895 FORMAT(1X)
C  RETURN TO MAIN FOR CALL VIA TO NAMED SUBROUTINE IN ANY OVERLAY
C  RETURN
END
OAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

(NOT IMPLEMENTED)
(NOT IMPLEMENTED)
SUBROUTINE PLCHAP( & INITIATE PLOTTING OF CLASSIFICATION MAP(S) 
U KOND) & I: FIRST 3 CHAR OF COMMAND O: SPACES

HISTORY
R E HARVESON LEC 12/10/78 MODIFIED FROM PRCHAP
E H SCHLOSSER LEC 01/15/79 REVISE & DOCUMENT

METHOD
????

MACHINE-DEPENDENT CODE
NONE.

EXTERNAL REFERENCES
GETCN    GET/CHECK INTEGER FIELD FROM UNIT S
WARN      OUTPUT WARNING DIAGNOSTIC FOR PREVIOUS FIELD FROM UNIT S
MDWARN    OUTPUT WARNING DIAGNOSTIC
CALSYL    CALIBRATE SYMBOL TABLE FOR PLOTTING
CALSCA    CALIBRATE PPD TRANSFORMATIONS FOR SCALE & DEVICE
CALWIN    CALIBRATE WINDOW ENVELOPES & VERTICES
OPENPL    OPEN PLOT FILE
MAPLOT    GENERATE MAP ON PLOTTER
SUBWIN    BREAK WINDOW INTO SUBWINDOW MAPS. BASED ON PRIMARY TICKS
MOCLRW    CLEAR COUNT OF 'WARNING' DIAGNOSTICS

EXCEPTIONS
1. ?????

GLOBAL DECLARATIONS

INCLUDE KOMXOT.LIST  COMMON PROGRAM EXECUTION SWITHCES. COUNTERS
INCLUDE WINDEF.LIST   DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE KOMINW.LIST   COMMON INPUT WINDOW PACKETS
INCLUDE KOMOUTW.LIST  COMMON OUTPUT WINDOW PACKETS
INCLUDE NULCST.LIST   DEFINE NULL CHARACTER STRING
EXTERNAL MAPLOT   GENERATE MAP ON PLOTTER

L-381
C LOCAL DECLARATIONS
C
INTEGER NSUBW  8 MAXIMUM NUMBER OF SUBWINOWS
INTEGER TEMP  8 TEMPORARY INTEGER
C
C PROCEDURE
C---------
C
C CALL TRACE
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 CALIBRATE/CHECK SPECIFICATIONS
C
IF(NWNDOW.EQ.0) CALL WARN3('INVALID DEFAULT COMMAND --')
CALL GETSIN(ITEMP, +1,-1,'EXTRA MAP SPECIFICATION --')
CALL CALSYL
CALL CALSCA(FLOAT(LINCH),FLOAT(KINCH)) 8=8=8=8=8=8=8=8=8=8=8
CALL CALWIN(0.25)
IF(MCHECK.NE.0) GO TO 900
C
C CHECK IF ANY DATA LIES WITHIN OUTPUT WINDOW
C
IF(MSAOWW(WLIN.WMAX).GE.MSAIWW(WLIN.WMAX)) OR.
& (MSAOWW(WLIN.WMAX).LE.MSAIWW(WLIN.WMIN)) OR.
& (MSAOWW(WSAM.WMIN).GE.MSAIWW(WSAM.WMAX)) OR.
& (MSAOWW(WSAM.WMAX).LE.MSAIWW(WSAM.WMIN)) CALL MDWARN
& 'NO MSS DATA WITHIN WINDOW'
C
C OPEN PLOT FILE: IF NOT OPEN, CLEAR WINDOW NUMBER & RESET PAGE NUMBER
C
IF(NOTOLT.NE.0) GO TO 900
IF(NWNDOW.LT.0) CALL OPENPL 8 OPEN PLOT FILE BEFORE 1ST WINDOW
NWNDOW=ABS(NWNDOW)
NPAGE=0
C
C GENERATE WINDOW MAP
C
IF(NSUBW.NE.0) GO TO 400
CALL MAPLOT
GO TO 900
C

L-382
! PLCHAP

DAW PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

 C GENERATE GEOGRAPHIC SUBWINDOW MAPS
 C
 400 IF((KSYSN(0'TIC), NE., 'DEO').AND.
  6 (KSYSN(0'TIC), NE., 'MIN')) GO TO 600  & NOT GEOGRAPHIC
   CALL SUBWIN(GEODOMN, NSUBW, MAPLOT)
   GO TO 900

 C GENERATE UTM SUBWINDOW MAPS
 C
 600 IF((KSYSN(0'TIC), NE., 'KM').AND.
  6 (KSYSN(0'TIC), NE., 'MET')) GO TO 800  & NOT UTM
   CALL SUBWIN(UTMDOMN, NSUBW, MAPLOT)
   GO TO 900

 C INVALID TICK INTERVAL FOR GENERATING SUBWINDOW MAPS
 C
 800 CALL MONOTE( 'SUBWINDOWS NOT ALLOWED FOR CURRENT PRIMARY TICKS')

 C ANY DIAGNOSTICS???
 C
 900 IF(NDLAFATL.EQ.0) GO TO 920
      CALL MONOTE( 'PREVIOUS FATAL ERRORS -- NO MAP GENERATED')
      GO TO 990
 920 IF(NNOWARN.EQ.0) GO TO 990
      CALL MONOTE( 'PREVIOUS WARNINGS -- NO MAP GENERATED')
      IF(NMBATC.EQ.0) WRITE(6,925)
      925 FORMAT(9X,'**TRY AGAIN!!')
      CALL MCCLRW( NULCS1)
 990 WRITE(6,995)
 995 FORMAT(9X,995)
      KOMD="
      RETURN
 END
SUBROUTINE PLCXGT 8 INITIALIZATION ROUTINE FOR PLTCCLASS

(R E HARVESEN)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

PSTART
NVIAO
ERCSF
OPNIEN
CLSHDD

INCLUDE KONQT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
EXTERNAL PLC000.NULSUB

IDENTIFY PROGRAM

CALL PSTART( 'DAM PLTCCLASS(7903)')

ON RETURN. CALL PLC000 TO GET DEFAULT/USER COMMANDS

CALL NVIAO( PLC000.NULSUB)

OPEN DETECTION FILES AND IDENTIFY ERTS SCENE

CALL OPNIEN
CALL CLSHDD( 8)

QUEUE DEFAULT COMMANDS FROM PRIVATE PROGRAM FILE OR DAM PROGRAM FILE

300 CALL SYSADD(LOCFILE, 'MACDAM','DEF-PLTCCLASS','')
 IF (LOCFILE.LE.0) CALL SYSADD(LOCFILE, 'DAM','DEF-PLTCCLASS','')
 IF (LOCFILE.LE.0) CALL MSATL( 'NO DEFAULT COMMANDS')

RETURN TO NEXT STATEMENT IN CALLING ROUTINE
RETURN
END
NOT IMPLEMENTED
NOT IMPLEMENTED
(NOT IMPLEMENTED)
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

(NOT IMPLEMENTED)
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

(NOT IMPLEMENTED)
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

PROGRAM FLNCCLASS NOT IMPLEMENTED
PROGRAM STATUS
---------------

(E H SCHLOSSER)

This program allows a user of the DAM package to determine the status of
specified batch or demand runs.

To retrieve summary information on a run, enter the runid. Transparent
characters (/) may be imbedded in the query runid to cause the retrieval of
summary information on a selected series of runs.

To retrieve detailed information on the programs executed within a run,
enter the run index (displayed in the summary information).

To terminate program execution enter:
EXIT

EXTERNAL SUBROUTINES/FUNCTIONS CALLED
----------------------------------------

PSTART
ERCSF
MLOG
READS
ERION
PSTOP

COMMON
1 LOPKT (6).
2 LOMOR.LOPL0.LOPHI.LORHI.LOMAX.  a I/O PACKET FOR LOG FILE
3 NCUEINDEX.LORNEW.
4 LOROLD.FUTURE (2).LPCT (25).
5 LPIDT (15).XX (13).
6 LDLGN (15).YY (13).
7 LTERM (14).ZZ (01).LSUPS
DATA LOPKT /'1','4*0/:
INCLUDE KOMIO.LIST
INCLUDE NULCST.LIST
DIMENSION MSOCUE(15)
DIMENSION HSNONE(')
DATA MSOCUE/
6 'ENTER RUNID'.NULCST.
6 'ENTER RUNID FIRST TO FIND INDEX'.NULCST.
6 'ENTRA RUNID OR INDX'.NULCST/
DIMENSION HSNONE(')
DATA HSNONE/' RUN ?????? NOT IN LOG'.NULCST/

CALL PSTART( 'DAM STATUS(80091')
NCUE-15  a NO CUE MESSAGE
NACCT-19  a DON'T PRINT ACCOUNT NUMBERS

L-391
NRULIM=20  8 MAX NUMBER OF RUNS LISTED FOR NON-PRIVILEGED USER
CALL ERCSF(ISTAT,'BAOD DAN.DEF-STATUS .')

C
C READ COMMAND FROM UNIT 5 (CARD READER OR TERMINAL)
C
300 KOMD=''
   CALL READS(ILSTAT,MSQCUC(NCUE)) 8
   IF(ILSTAT.NE.' ') KOMD='EXIT'
   IF(KOMD.EQ.' ') CALL GETSKN(KOMD,(B),NULCST) 8 GET 8 ALPHA CHARS

C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALENT
C
360 KASE-ICE(KOMD)-ICE('A')+1 0 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 0 NOT ALPHA
   GO TO 1
   1 411.412.413.414.415.416.417.418.419.420.
   & .KASE

C
C DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOMD TO BLANK
C
401 CONTINUE 8*** A
402 CONTINUE 8*** B
403 CONTINUE 8*** C
404 CONTINUE 8*** D
   GO TO 500

405 CONTINUE 8*** E
   IF(KOMD.EQ.'EXIT') OR.
   & (KOMD.EQ.'EXIT') CALL PSTOP(INULCST)
   IF(KOMD.EQ.'EXPLA') OR.
   & (KOMD.EQ.'EXPLA') OR.
   & (KOMD.EQ.'EXPL') OR.
   & (KOMD.EQ.'EXP') CALL KMDEXP(KOMD)
   GO TO 500

406 CONTINUE 8*** F
407 CONTINUE 8*** G
408 CONTINUE 8*** H
409 CONTINUE 8*** I
410 CONTINUE 8*** J
   GO TO 500

411 CONTINUE 8*** K
   IF(KOMD.EQ.'KEY') CALL STAAKEY(KOMD)
   GO TO 500

412 CONTINUE 8*** L

L-396
413 CONTINUE 8*** M
   GO TO 500

C 414 CONTINUE 8*** M
   IF(KOMO.EQ.'NEWS' ) OR.
   & (KOMO.EQ.'NEW ' ) CALL KMDNEW(KOMO)
   & (KOMO.EQ.'NEXT ' ) OR.
   & (KOMO.EQ.'NEXT' ) CALL KMDNEX(KOMO)
   30 TO 500

C CONTINUE 8*** N
   IF(KOMO.EQ.'OFF ' ) NCUE=1  & ASK FOR RUNID AFTER DEFAULT COMMANDS
   IF(KOMO.EQ.'OFF ' ) CALL KMOOFF(KOMO)
   IF(KOMO.EQ.'ON ' ) CALL KMODON(KOMO)
   00 TO 500

C 416 CONTINUE 8*** P
417 CONTINUE 8*** Q
418 CONTINUE 8*** R
419 CONTINUE 8*** S
420 CONTINUE 8*** T
421 CONTINUE 8*** U
422 CONTINUE 8*** V
423 CONTINUE 8*** W
424 CONTINUE 8*** X
425 CONTINUE 8*** Y
426 CONTINUE 8*** Z
   GO TO 500

C 427 CONTINUE 8*** NOT ALPHABETIC
   IF(KOMO.EQ. ' ' ) GO TO 300  & IGNORE BLANKS
   GO TO 600

C C LIST RUN SUMMARY
C 500 IF(KOMO.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 LIST PROGRAM EXECUTION SUMMARY
C 600 CALL UNGETS
   LINDEX=999999
   CALL GETSIN(LINDEX, 999999,NULCST)
620 IF(LINDEX.EQ.999999) GO TO 500  & NOT A VALID INDEX. MUST BE RUNID:
   IF(ILOMAX.NE.0) GO TO 640
   NCUE=4  & ASK FOR RUNID, NOT INDEX!
   GO TO 300
   640 CALL GETXOT
   IF(LINDEX.EQ.0) GO TO 300
   CALL READS(LSSTAT, ' ' )
SUBROUTINE STAKEY(KOMO)
  CALL GETSKH(KEY, NULCST)
  CALL VALKEY(KEY)
  NACCT = 21 8 PRINT ACCOUNT NUMBERS
  NRULIM = 99999 8 ALLOW ANY NUMBER OF RUNS TO BE LISTED
  WRITE(8, 105) 105 FORMAT("**DESTROY*SECURITY*KEY**")
  KOMO = ' ' RETURN

SUBROUTINE OETRUN

COUNT TRANSPARENT (/) RUNID CHARACTERS

DO 150 NCHAR = 1, 6
  CALL OETCHR(KHAR, KOMO, NCHAR)
  IF(KHAR.EQ.' ') NTRNSP = NTRNSP + 1
150 CONTINUE
  NRUNMAX = NTRNSP + 2
  CALL OETSIN(NRUNMAX, 1, NRULIM, 'BAD MAXIMUM RUN NUMBER --')
  CALL OETSIN(ITEMP, +1, 1, 'EXTRA RUN SPECIFICATION --')
  IF(NTRNSP.NE.0) GO TO 170 8 ONE OR MORE TRANSPARENT CHARS

CHECK RUNID FOR INVALID CHARACTERS

DO 180 NCHAR = 1, 6
  CALL OETICE(NICE, KOMO, NCHAR) 8 GET INTEGER-CHARACTER-EQUIVALENT
  IF(6 (NICE.LT.ICE('A').OR.NICE.GT.ICE('Z')) .AND.
     6 (NICE.LT.ICE('0').OR.NICE.GT.ICE('9'))
     6 1 GO TO 170 8 INVALID CHARACTER

L-394
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

160 CONTINUE

C CHECK IF QUEUED READXX CARD INPUT FILE EXISTS FOR SPECIFIED RUNID

JUSER(I) = JASOR  & RUNID
CALL ERCSF(ISTAT,JUSER)
CALL ERCSF(ISTAT,JASOR)
IF (ACCEPT(ISTAT).EQ.0) GO TO 170  ! NO READXX QUEUED CARD INPUT FILE

C IDENTIFY QUEUED RUN AS BACKLOG OR ACTIVE

LORNEW = -999999
LPCT(I) = JASOR  & RUNID
FLO(IOB,B,LPC;l(25)) = 0 & 'BATCH'
LPCT(I) = '999888' & DATE = 00/00/00  & TIME = 00:00
IF (LOTHRUN(ISTAT).EQ.0) GO TO 165  & NOT YET BEING USED
LPCT(I) = '(ACTIVE'
LPCT(I) = 'E'
GO TO 167
165 LPCT(I) = '(BACKL'
LPCT(I) = '001'
167 CALL ERCSF(ISTAT,JFREE)
CALL PRTRUN

C READ LOG FILE HEADER

170 IOSIZE(LOGKT) = 5
IOADDR(LOGKT) = LOC(LOGHDR)
IOSECT(LOGKT) = 0
IOFUNC(LOGKT) = '8K'  & READ
CALL ERIOW(LOGKT)
LORNEW = LORHI

C CHAIN READ RUN HEADERS IN REVERSE ORDER

180 IOSIZE(LOGKT) = 28
IOADDR(LOGKT) = LOC(LOGROLD)
IOSECT(LOGKT) = LORNEW
IOFUNC(LOGKT) = '8K'  & READ
CALL ERIOW(LOGKT)
IF (MATCHR(JASOR,LPCT(I)).GE.(6-NTRNSP)) CALL PRTRUN
IF (NRUNS.LE.NRUMAX) GO TO 900
IF (LOGOLD.LT.4) GO TO 900
IF (LOGNEW.LE.LOPLO) AND
1 (LRGOLD.LE.LOPLO) GO TO 900
LORNEW = LOROLD
GO TO 200

C CHECK IF ANY SPECIFIED RUNS WERE FOUND

900 IF (NRUNS.NE.0) GO TO 990

L-395
FUNCTION MATCHR(JWD1, JWD2)

MATCHR = 0
DO 150 NBIT = 0, 30, 6
  IF (FLD(NBIT, 6, JWD1).EQ.FLD(NBIT, 6, JWD2)) MATCHR = MATCHR + 1
150 CONTINUE
RETURN

SUBROUTINE PTRUN

DIMENSION MODE(6)
DATA MODE/ 'MIEXEC', 'RLTIME', 'LOEXEC', 'DEMAND', 'DEADLN', 'BATCH'/

IF (NRUNS.EQ.0) WRITE(6, 135)
135 FORMAT('0'/' INDEX RUN NO MODE '.,
  1 'DATE ', TIME ',
  2 'QUALIFIER ', ACCOUNT'/'IX)
LTYP = FLD(06, 6, LPC(25))
LMON = FLD(00, 6, LPC(15))
LDAY = FLD(08, 6, LPC(15))
LYR = FLD(12, 5, LPC(15)) + 1960
LHR = (FLD(19, 18, LPC(15)) - 3600) / 3600
LMIN = FLD(18, 18, LPC(15)) - 3600 * LHR / 60
WRITE(6, 155) LYP, NEW, LPC(N), MODE(LYP),
  1 LMON, LDAY, LHR, LMIN.
2 (LPC(N), N=18, NACC)
155 FORMAT(16, 2X, 46.2X, 46.2X).
  2 'J2', 'J2'.
NRUNS = NRUNS + 1
RETURN

SUBROUTINE GETXQ

DIMENSION LQUAL(2), MDAYS(12)

MONTH: 01 02 03 04 05 06 07 08 09 10 11 12

L-396
DATA MDAYS/31,28,31,30,31,30,31,30,31,30,31,31/
C
C READ LOG SECTORS FOR FIRST PROGRAM EXECUTION OF RUN
C 110 LSECTR=INDEX
INDEX=0
IF(LSECTR.LT.4) GO TO 810
IF(LSECTR.GT.LMAX) GO TO 810
IF(MOD(LSECTR,4).EQ.0) GO TO 810
IOSIZE(LOPKT)=100   # 3 SECTORS & FIRST 16 WORDS OF NEXT SECTOR
I0SECT(LOPKT)=LSECTR
IOFUC(LOPKT)='SK'    # READ
CALL ERIOQ(LOPKT)
IF(IOSTAT(LOPKT).NE.0) GO TO 810
IF(MATCHR(LPCT(1),LPCT(2)),LT.5) GO TO 810   # ORIG/GEN RUNID NOT SIMILAR
LYR=FLD(06,6,LPCT(25))
IF(KMOND.EQ.1) GO TO 310   # MONITOR ACTIVE PROGRAM
IF(LORDLD.EQ.1) CALL WARN:('PARTIAL RUN INDEX ----')
LRORIO=LPCT(1)
LRGEN=LPCT(2)
LMYDS=LPCT(15)
LMYR=FLD(12,6,LPCT(15))
MDAYS(2)=29
IF(MOD(LYR,4).EQ.0) MDAYS(2)=29   # LEAP YEAR!
QAL(1)=LPCT(10)
QAL(2)=LPCT(19)
WRITE(6,125) LSECTR,LRORIO,LRGEN,QAL
125 FORMAT('0'/'(',J5,')' ',16,')/' ',10,') 2A6/IX.4(',-----')
GO TO 310
C
C READ LOG SECTORS FOR SUBSEQUENT PROGRAM EXECUTIONS
C 200 I0SECT(LOPKT)=LSECTR
IOFUC(LOPKT)='SK'    # READ
CALL ERIOQ(LOPKT)
IF(IOSTAT(LOPKT).NE.0) GO TO 900
IF(MATCHR(LPCT(1),LPCT(2)),LT.5) GO TO 900
IF(FLD(06,6,LPCT(15)).LT.6) AND.
IF((FLD(06,6,LPCT(1)) - MOD(LOAD,MDAYS(LMON)))).GT.1) GO TO 900
IF(LPCT(2),NE.,LPCT(19)) GO TO 400
IF(LPCT(19).NE.,LMYDS) GO TO 400
C
C WRITE QUALIFIER IF CHANGED
C 300 INDEX=0
IF((QAL(1).EQ.,LPCT(18))).AND.
IF((QAL(2).EQ.,LPCT(19))) GO TO 310
QAL(1)=LPCT(18)
QAL(2)=LPCT(19)
WRITE(6,305) QAL
C
L-397
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/RoutineS

305 FORMAT(' QUALIFIER CHANGED TO ',2A8/1X,4('------'))

WRITE PROGRAM ID/DATE/TIME & DIAGNOSTIC/TERMINATION LOG ENTRIES

310 WRITE(6,315) LPI0T
315 FORMAT(1X,15A8)
   IF(LOIAON(1),NE,' ') WRITE(6,325) LOIAON
325 FORMAT(2X,14A8)
   WRITE(6,335) LTERM
335 FORMAT(1X,14A8/1X,4('------'))

CHECK PROGRAM TERMINATION STATUS

IF(LTERM(1),EQ,' NORMA'),OR.
   & (LTERM(1),EQ,' ERROR') GO TO 400
   IF(LTERM(1),NE,' ABOR'), GO TO 360
   IF(LTYP.EQ,4) GO TO 400 & IN DEMAND ABORT DOES NOT KILL RUN
   GO TO 900

INCOMPLETE PROGRAM EXECUTION -- SAVE SECTOR POINTER

380 LINDEX=LSECR

TEST AND INCREMENT LOG INDEX

400 IF(LSECR.EQ.LOPH1) GO TO 900
   LSECR=MOD(LSECR,LOMAX)+1
   GO TO 200

FLAG INVALID INDEX

810 CALL MDWARN('INVALID INDEX--')
   WRITE(6,815) LSECR
815 FORMAT(6X,16)

900 RETURN
END
PROGRAM STATUS/VIRTUAL

HISTORY

E N SCHLOSSER  LEC  08/02/74  ORIGINAL CODE
E N SCHLOSSER  LEC  11/06/79  SHAP.FZINIT: 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 6-BIT FIELD DATA CHARACTERS.
IMPLEMENTING CODE MUST BE REWRITTEN FOR DIFFERENT CHARACTER CODES,
DIFFERENT OPERATING SYSTEMS, AND DIFFERENT MACHINES.

EXTERNAL REFERENCES

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

EXCEPTIONS

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

GLOBAL DECLARATIONS

PROGRAM TYPE IS PRE-LOADED BY EXEC INTO REGISTER 44 AS FOLLOWS:
I 2 = REAL TIME
I 3 = LOW EXEC
I 4 = DEMAND
I 5 = DEADLINE BATCH
I 6 = BATCH

SXQT OPTIONS ARE PRE-LOADED BY EXEC INTO REGISTER AS IN
MASTER BIT NOTATION.

LOCAL DECLARATIONS

AXRS

$001 D-BANK
$SSH FORM 9.9.9.19

LASSDF $SSH 030.1.F'-0 LABEL: I NO. FORTRAN FIELD DATA

LASSMO *SOFF

NAPDF $SSH 000.9.9.0 DATA: 9 WDS. FIELD DATA

NAPIMO '8XQTS: MAP,FZN DAM,STATUS-MAP STATUS : 8XQTS'

ADDSDF $SSH 000.9.9.0

ADDOM '8XQTS: ADD DAM, SYS-MAPOPT : 8XQTS'

XQTSDF $SSH 000.9.9.0

XQTIMO '8XQTS: XQT: STATUS : 8XQTS'

EOFSDF 0 END-OF-FILE STOP WORD

PF FORM 9.9.9.19

CSFASO '8ASO,T 20.

CSFADD '9ADD 20.

SAVREG RES 1

IOPKT 1000 '20.WS 33.LASSDF,'0' 0

PROCEDURE

$001 I-BANK

STATUS* LA.U A0.  ' ' A0 = ' ' ' ' TNE.U A4.4 SKIP NEXT INST IF A4<4 (NOT DEMAND)

S6.SZ LA.0.MAPIMO+Z DEMAND BLANK OUT N OPTION

LA A0.(CSFASQ) ADDRESS OF BASO IMAGE

ER CSFS DO IT

SA A0.SAVREG STORE &

PSPRINT (PF 2.1.SAVREG) PRINT BASO STATUS

GETOPT LOAD OPT LTRS INTO A2.A3.A4

PUTOPT DS A2.XQTIMO+2 STORE OPTION LETTERS INTO XQT IMAGE

SA A4.XQTIMO+4 (3 WORDS -- MAX 18 OPT LETTERS)

WRITE LA A0.(IOPKT) ADDRESS OF I/O PACKET

ER IOMS WRITE SDI IMAGES TO 20.

ADD LA A0.(CSFADD) ADDRESS OF ADD IMAGE

ER CSFS DO IT

ER EXITS

END STATUS

L-401
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAN.STATUS/..NTABS/DAN..SYS-BLOCK
LIB DAN.
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM STATUS/VIRTUAL

STATUS-MAP/VIRTUAL 081
PROGRAM DITCOP

(J C POOLEY)

THIS PROGRAM COPIES *DAMDET*(1-4) DISC DETECTION FILES TO TAPE.

INPUT CONSISTS OF *DAMDET*(1-4) DETECT DISC FILES.
OUTPUT CONSISTS OF ONE TO FOUR FILES ON A SINGLE REEL OF TAPE.
THE DISC FORMAT IS DOCUMENTED IN DAM PACKAGE CLASSIFY.
THIS TAPE FILE IS IN UNIVAC EXEC 8 COPY FORMAT.

INCLUDE KOMXOT
INCLUDE KOMLOG
INCLUDE KONLUS
INCLUDE KONNER
INCLUDE KONFIT

EXTERNAL DIT000,DITXOT

CALL NVIATO DIT000,DITXOT) 8 FIRST CALL IS VIA DIT000 TO DITXOT
100 CALL VIATO
   GO TO 100
   END
PROGRAM DITCOP/VIRTUAL

HISTORY

E H SCHLOSSER LEC 07/07/78 ORIGINAL CODE
E H SCHLOSSER LEC 11/06/79 SNMP.FZIN1; NO 'N' IN DEMAND

METHOD

CONSTRUCT SNMP EXEC COMMAND TO LINK REAL ABSOLUTE IN TPFS.
CONSTRUCT $XOT COMMAND TO EXECUTE REAL ABSOLUTE IN TPFS.
WRITE SNMP & $XOT 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-9 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-9 CONTROL STATEMENT
ER IOWS $ INITIATE I/O AND WAIT FOR COMPLETION
ER EXITS $ TERMINATE PROGRAM EXECUTION
DAM.DITCOP-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 & 8PREP-D.

GLOBAL DECLARATIONS

1. 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

18XOT OPTIONS ARE PRE-LOADED BY EXEC INTO REGISTER AS IN
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

**MASTERS BIT NOTATION.**

**LOCAL DECLARATIONS**

```
AXRS
$00 . D-BANK
SSH FORM 6.6.6.10
   111111122222333334444455555666667777788888999999
LABSDF SSH 030.1. 'F'. YouTube FORTRAN FIELDATA
LABINO SSH '43DFE'
MAPSDF SSH 000.9.0.0 . DATA. 9 WDS. FIELDATA
MAPINO '8QTS: DITCOP-HAP-DITCOP '. 8QTS'
ADDOS SSH 000.9.0.0
ADDINO '8QTS: ADD DAM.SYS-AMOPE'. '8QTS'
XQTSDF SSH 000.9.0.0
XQTSNO '8QTS: XOT.1'. DITCOP '. 8QTS'
EOFSDF - 0 . END-OF-FILE STOP WORD
PF FORM 12.8.18
CSFASO '8ASG.T 20. '.
CSFADD '8ADD 20. '
SAYREG RES 1
IOPKTI ISOD '20'.MS 33.LABSDF. '0' 0

PROCEDURE

```

```
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM.DITCOP/...NTABS/DAM...SYS-BLOCK
LIB DAM.
DAM PACKAGE APPENDIX L
MAIN PROGRAMS/ROUTINES

IN DAM.DITCOP/VIRTUAL
SUBROUTINE DIT000: A CALL PHASE 0 SUBROUTINES FOR DITCOP
& NAMSUB) A NAME OF SUBROUTINE TO CALL (OR NULSUB)

HISTORY

E H SCHLOSSER  LEC  07/07/78  ORIGINAL CODE
E H SCHLOSSER  LEC  01/19/79  MACRO COMMANDS & TIME COMMAND

METHOD

NEXT COMMAND IS RETRIEVED, VALIDATED, AND ITS SUBROUTINE CALLED.

EXTERNAL REFERENCES

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

EXCEPTIONS

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

GLOBAL DECLARATIONS

INCLUDE NULCST.LIST  A DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

INTEGER KMD  A FIRST 3 CHARS OF USER COMMAND (BLANK AFTER DONE
INTEGER LSSTAT  A 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 NAMSUB  # 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, NULCST)  # FILL UNIT 5 BUFFER, NO CUE MESSAGE
IF (LSSTAT.NE.'I') KOMD='EXI'
IF (KOMD.NE.'EXI') CALL GETSAL(KOMD,(3), NULCST)  # GET 3 ALPHA CHAR
C
C CONVERT FIRST CHARACTER OF COMMAND TO INTEGER-CHARACTER-EQUIVALEN
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.11) OR (KASE.GT.26) KASE=27  # NOT ALPHA
00 TO
0 401, 402, 403, 404, 405, 406, 407, 408, 409, 410.
1 411, 412, 413, 414, 415, 416, 417, 418, 419, 420.
6 .KASE
C
C DETERMINE COMMAND, PERFORM COMMAND, CHANGE KOMD TO BLANK

# CONTINUE:

401 CONTINUE #**** A
00 TO 800
C
402 CONTINUE #**** B
00 TO 800
C
403 CONTINUE #**** C
IF (KOMD.EQ. 'CLE') CALL KMDCLE(KOMD)    # CLEAR
00 TO 800
C
404 CONTINUE #**** D
IF (KOMD.EQ. 'DUP') CALL DITDUP(KOMD)
00 TO 800
C
405 CONTINUE #**** E
IF (KOMD.EQ. 'EXI') CALL DITEXI(KOMD)    # EXI
IF (KOMD.EQ. 'EXP') CALL KMDEXP(KOMD)    # EXPLAIN
00 TO 800
C 408 CONTINUE 8**** F
C 407 CONTINUE 8**** G
C 406 CONTINUE 8**** H
C 409 CONTINUE 8**** I
C 410 CONTINUE 8**** J
C 411 CONTINUE 8**** K
C 412 CONTINUE 8**** L
C 413 CONTINUE 8**** M
C 414 CONTINUE 8**** N
C 400 TO 000
C 407 CONTINUE 4000 0
C 409 CONTINUE 4000 1
C 410 CONTINUE 4000 2
C 411 CONTINUE 4000 3
C 418 CONTINUE 4000 4
C 413 CONTINUE 4000 5
C 414 CONTINUE 4000 6
C 415 CONTINUE 8**** O
C 400 TO 000
C 407 CONTINUE 4000 0
C 409 CONTINUE 4000 1
C 410 CONTINUE 4000 2
C 411 CONTINUE 4000 3
C 418 CONTINUE 4000 4
C 413 CONTINUE 4000 5
C 414 CONTINUE 4000 6
C 415 CONTINUE 8**** P
C 416 CONTINUE 8**** Q
C 417 CONTINUE 8**** R
C 418 CONTINUE 8**** S
C 419 CONTINUE 8**** T
C 420 CONTINUE 8**** U
C 421 CONTINUE 8**** V
C 422 CONTINUE 8**** W
C 423 CONTINUE 8**** X
C 424 CONTINUE 8**** Y
C 425 CONTINUE 8**** Z
C 426 CONTINUE 8**** NOT ALPHABETIC
C C IF COMMAND WAS NOT FOUND, TRY MACRO-COMMAND
C C 800 IF(KOMD.NE.' ') KOMD='DIT--' A 1ST 3 CHARS OF PROG NAME PLUS '--'
C C 800 IF(KOMD.NE.' ') CALL KMDXXX(KOMD) A MACRO COMMAND HANDLER
C C COMMAND IS INVALID IF STILL NOT FOUND
C C IF(KOMD.NE.' ') CALL WARNS('INVALID COMMAND --')
C C RETURN TO MAIN FOR CALL VIA/TO NAMED SUBROUTINE IN ANY OVERLAY
C C RETURN
C END

ORIGINAL PAGE IS OF POOR QUALITY.
SUBROUTINE OITEXI & TERMINATION ROUTINE FOR OITCOP
---------------------------------------------------

(E H SCHLOSSER)

INCLUDE NMXQOT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST & DEFINE NULL CHARACTER STRING

CALL TRACE

EXTERNAL SUBROUTINES/FUNCTIONS CALLED
----------------------------------------

PSTOP

TERMINATE PROGRAM

IF (NDFAFL .NE. 0) CALL PABORT(NULCST)
CALL PSTOP(NULCST)

PSTOP DOES NOT RETURN

END
SUBROUTINE DITDUP1 8 COPY DISC TO TAPE ROUTINE
0 KOND1 8 NORMAL RETURN

HISTORY

J C POOLEY  LEC 01/03/80  ORIGINAL CODE
CHARLES HELMKE  LEC 01/03/80  ORIGINAL CODE

METHOD

VERIFY THAT A TAPE IS ASSIGNED TO UNIT 2. WRITE KOMDET COMMON
BLOCK AS A HEADER FILE ON TAPE. USE SCRATCH FILE TO BUILD
A SERIES OF COMMANDS TO COPY EXISTING DETECTION FILES TO TAPE.
ADD SCRATCH FILE TO RUNSTREAM.

MACHINE-DEPENDENT CODE

EXTERNAL REFERENCES

ERCSF 8 SUBMIT EXEC-0 CONTROL STATEMENT FUNCTION
FLINFO 8 GETS FILE INFORMATION
MNOTE 8 PRINT/LOG/COUNT 'NOTE' MESSAGES
MWARN 8 PRINT/LOG/COUNT 'WARNING' MESSAGES
MFATL 8 PRINT/LOG/COUNT 'FATAL ERROR' MESSAGES

EXCEPTIONS

THE FOLLOWING CONDITIONS GENERATE THE DIAGNOSTICS SHOWN:

CONDITION  DIAGNOSTIC
NO TAPE ASSIGNED  FATAL MESSAGE
PREVIOUS WARNING MESSAGE(S)  NOTE MESSAGE
PREVIOUS FATAL MESSAGE(S)  NOTE MESSAGE
IN CHECKOUT MODE  NOTE MESSAGE

GLOBAL DECLARATIONS

INCLUDE KOMXQT.LIST  8 COMMON PROGRAM EXECUTION SWITCHES.COUNTERS
INCLUDE KOMIO.LIST  8 FORTRAN MANIPULATION OF ASSEMBLER I/O PKT
INCLUDE KOMDET.LIST  8 WINDOWS & GENERATION DATES FOR DET FILES
INCLUDE ASHDEF.LIST  8 UNIVAC 1100 ASSEMBLER PARTIAL WORD
INCLUDE KOMLEN.LIST  8 I/O PACKETS FOR DETECTION FILES
INCLUDE KOMLUS.LIST  8 COMMON BUFFER.POINTERS.FLAGS FOR UNIT 8
INCLUDE FIDDEF.LIST  8 DEFINES RECORD STRUCTURE

C LOCAL DECLARATIONS
C --------------------------------------------------
C
INTEGER IDFILE(10)  8 ARRAY IN FIDDEF FORMAT
INTEGER
8 JASOTE(4) '/BASQ.T 20.' FILASQ '/

C PROCEDURE
C -----------
C
CALL TRACE
C
CALL ENCSF(NAO,JASOT)
C
CALL FACILITIES ASSIGNMENT FOR UNIT 2
C
CALL FLINFO(IDFILE , '2', '80')
1 IF(IDFILE(FIOGT).NE. 'TAPE')
2 CALL NDFATL('TAPE NOT ASSIGNED TO UNIT 2')
1 IF(NDTOTL.NE.0) GO TO 700
C
CALL WRITE TOC FILE TO TAPE
C
10 FUNC(LUEPRT)= '8C'
10 SIZE(LUEPRT)= SIZEFT
10 ADDR(LUEPRT)=LOC(KOMDET)
10 WAIT(LUEPRT)= 30
C
CALL ENQON(LUEPRT)
C
CALL WRITE END OF FILE TO TAPE
C
10 FUNC(LUEPRT)= '80'
C
CALL ENQON(LUEPRT)
C
C CHECK DATE WORD JENDMY(NCCT) OF DETECTION FILES
C
DO 600 NCCT=1,4
1 IF(JENDMY(NCCT),EQ. ' ') GO TO 600
C
L-414
DAN PACKAGE APPENDIX L
MAIN PROGRAMS/Routines

C *DAMDET-(NCCT) CATALOGED
C
C WRITE(20,305) NCCT
C 305 FORMAT('COPY OF +DAMDET=11...2')
C
C END OF LOOP VERIFYING ASSIGNMENT
C
C 600 CONTINUE
C
C WRITE MESSAGE TO REWIND TAPE FILE 2.
C
C 700 WRITE(20,705)
C 705 FORMAT('REQUEST TO END PROGRAM/0** PLEASE FREE 2. OR REWIND 2./*
C       '1:SEOF')
C
C ENDFILE 20 TO DRAIN BUFFER
C
C ENDFILE 20
C
C ADD FILE 20 TO RUNSTREAM
C
C CALL ERC3FNAO.'SADD 20. .COPY FILE '
C
C CHECK DIAGNOSTIC COUNTERS
C
C 800 IF(NOMARN.EQ.0) GO TO 820
C CALL MONOTE('PREVIOUS WARNINGS -- NO COPYING PERFORMED')
C IF(NBATCH.EQ.0) WRITE(6,815)
C 815 FORMAT('...TRY AGAIN')
C GO TO 900
C
C 820 IF(NOFATL.EQ.0) GO TO 850
C CALL MONOTE('PREVIOUS FATAL ERRORS -- NO COPYING PERFORMED')
C GO TO 900
C
C 850 IF(INECHECK.EQ.0) GO TO 900
C CALL MONOTE('CHECKOUT MODE -- NO COPYING PERFORMED')
C
C NORMAL RETURN
C
C 900 CONTINUE
C
C 900 CONTINUE
C
C RETURN
C
C END
SUBROUTINE DIVER1 & VERIFY DAMSEN TAPE COPY
& KOND1 & NORMAL RETURN

CALL NONOTE('VERIFY COMMAND NOT YET IMPLEMENTED')
SUBROUTINE DITKOT -- A INITIALIZATION ROUTINE FOR DITCOP

( J C POOLEY)

EXTERNAL SUBROUTINES/FUNCTIONS CALLED

CLSND0
ERCSF
OPN10N
PSTART
NVIATO

INCLUDE KUNNER
EXTERNAL DIT000.NULSUB

IDENTIFY PROGRAM

CALL PSTART("DAM DITCOP(7803)"

OPEN DENSITY FILES

CALL OPN10N

IDENTIFY ERFS SCENE

CALL CLSNDO(S)

ADD DEFAULT COMMANDS

300 CALL ERCSF(NAO,"8ADD DAM.DEF-DITCOP"

ON RETURN, CALL DIT000 TO GET DEFAULT/USER COMMANDS

CALL NVIATO(DIT000.NULSUB)

RETURN TO NEXT STATEMENT IN CALLING ROUTINE

RETURN

END
PREFACE TO APPENDIX M

THE SUBROUTINES IN THIS APPENDIX IMPLEMENT COMMANDS COMMON TO SEVERAL DAP 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. (KMDPOI & 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). KMDPOI & 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 (MCFRM = KMDPOI <> 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 O FOR DETAILED SYNTAX OF ALL COMMANDS.
ADJUST NETWORK (SEE APPENDIX L)
ALIGN COORDINATE SYSTEMS
GET/CHECK PLATFORM ATTITUDE
GET/CHECK SCENE CENTER SCAN COORDINATES
GET/CHECK RAM/TRANSFORMED CHANNEL(S)
GET/CHECK DETECTION CHANNEL(S)
CLEAR WARNINGS/ERRORS
GET/CHECK COLOR(S)
GET/CHECK NUMBER OF OUTPUT COPIES
GET/CHECK COUNT PER PIXEL
CROSSTABULATE
GET/CHECK DENSITY LIMITS
DETECT (SEE APPENDIX L)
DIAGRAM NETWORK (SEE APPENDIX L)
DISPLAY ON ALPHA-NUMERIC DEVICE (SEE APPENDIX L)
EXIT (SEE APPENDIX L)
EXPLAIN PROGRAM/COMMAND
FACTOR (SEE APPENDIX L)
END IF ... FI BLOCK
GET/CHECK SCENE GEOMETRY
GET/CHECK PAGE HEADINGS
HISTOGRAM (SEE APPENDIX L)
BEGIN IF ... FI BLOCK
GET/CHECK INTENSITY(S)
LIST (SEE APPENDIX L)
MAP (SEE APPENDIX L)
GET/CHECK TRANSVERSE MERCATOR CENTRAL MERIDIAN
MODEL (SEE APPENDIX L)
GET/CHECK MATERIAL NAME
PRINT NEWS OF PROGRAM CHANGES
SPECIFY CONDITION FOR PERFORMING NEXT COMMAND
TURN OFF MODE SWITCH(ES)
TURN ON MODE SWITCH(ES)
GET/CHECK WINDOW ORIGIN
PICTURE ON COLOR CRT (SEE APPENDIX L)
'POKE' (CHANGE) LABELLED COMMONS (FOR DEBUGGING)
GET/CHECK POLAR GAIN/BIAS
GET/CHECK PRINTER SPECIFICATIONS
PROFILE (SEE APPENDIX L)
PARTITION (SEE APPENDIX L)
'PEEK' INTO LABELLED COMMONS (FOR DEBUGGING)
GET/CHECK PLOTTER SPECIFICATIONS
GET/CHECK CONTROL/CHECK POINT
'POKE' (CHANGE) LABELLED COMMONS (FOR DEBUGGING)
GET/CHECK POLAR GAIN/BIAS
GET/CHECK PRINTER SPECIFICATIONS
PROFILE (SEE APPENDIX L)
PARTITION (SEE APPENDIX L)
'PEEK' INTO LABELLED COMMONS (FOR DEBUGGING)
GET/CHECK PLOTTER SPECIFICATIONS
GET/CHECK CONTROL/CHECK POINT
'POKE' (CHANGE) LABELLED COMMONS (FOR DEBUGGING)
GET/CHECK POLAR GAIN/BIAS
GET/CHECK PRINTER SPECIFICATIONS
PROFILE (SEE APPENDIX L)
PARTITION (SEE APPENDIX L)
G Package Appendix M
Command Routines

Command Routines

- GET/CHECK SHARPENING FILTER COEFFICIENTS
- GET/CHECK SCENE SIZE IN SCAN COORDINATES
- GET/CHECK WINDOW SPACING
- GET/CHECK SPHEROID GEOGRAPHIC PARAMETERS
- GET/CHECK SYMBOLE(S)
- TABULATE TICK INTERVALS
- PRINT CLOCK TIME & CHARGE TIME
- GET/CHECK TICK SPECIFICATIONS
- GET/CHECK TOLERANCE
- GET/CHECK WINDOW VERTICES
- GET/CHECK/PROCESS MACRO COMMAND (CALLS KMXXED)
- GET/CHECK UTM PROJECTION ZONE
- $ADD -- DYNAMIC $ADD
- $ASO -- DYNAMIC $ASO
- $BRKPT -- DYNAMIC $BRKPT
- $FREE -- DYNAMIC $FREE
- $LOG -- DYNAMIC $LOG
- EDIT ACTUAL SPECS INTO MACRO COMMAND DEFINITION
- GET/EVALUATE ACTUAL SPEC FOR MACRO COMMAND
SUBROUTINE KMDALII  & ALIGN COORDINATE SYSTEMS
U KOND) & I: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY

E H SCHLOSSER  LEC  10/30/75  ORIGINAL CODE
E H SCHLOSSER  LEC  07/17/78  UPGRADE DOCUMENTATION & DELETE RET K
E H SCHLOSSER  LEC  02/20/79  REVISE GETS/...U40/U4U CALLS
E H SCHLOSSER  LEC  11/07/79  REVISE 04A & GETSKH CALLS

METHOD

COMPUTE STM COORDINATES OF ALIGNMENT POINT FROM INPUT SCANNER
COORDINATES AND FROM INPUT EARTH COORDINATES. USE DELTAS BETWEEN
THESE COMPUTED STM COORDINATES TO MODIFY THE CURRENT TRANSFORMATION
COEFFICIENTS.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSXM  GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN  GET INTEGER DATA FIELD FROM UNIT 5
GETSXX  GET SEXAGENARY DATA FIELD FROM UNIT 5
GETSRL  GET REAL DATA FIELD FROM UNIT 5
WARNS  GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
U40  GEOGRAPHIC FOR UTM COORDINATES
REVERT  COMPUTE INVERSE LINEAR TRANSFORMATION COEFFICIENTS
U40  UTM FOR GEOGRAPHIC COORDINATES
MONOTE  PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
MDWARN  PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
04A  GEOGRAPHIC FOR ADJUSTED NSS COORDINATES

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   SCAINI SPECIFICATION MISSING OR MISSPELLED
   EARTH COORDINATE SYSTEM NOT DEFINED OR METRES) OR KM
   EARTH COORDINATE SYSTEM METRES) OR KM & CENTRAL MERIDIAN (ZONE)
   NOT DEFINED

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
GLOBAL DECLARATIONS


LOCAL DECLARATIONS

PROCEDURE

CALL TRACE

GET/CHECK SCANNER (OSFC-ADJUSTED) COORDINATES

GET/CHECK EARTH COORDINATE SYSTEM

M-5
KOROSV = 'NUL'
CALL GETSKH(KOROSV,(3), "NO EARTH COORDINATE SYSTEM --")
IF(KOROSV.EQ.'OE') GO TO 400
IF(KOROSV.EQ.'KM') GO TO 600
IF(KOROSV.EQ.'MET') GO TO 610
CALL WARS( 'BAD COORDINATE SYSTEM --')
GO TO 900

C
C GET/CHECK GEOGRAPHIC COORDINATES (DEGREES)
C
400 CALL GETSSX(OEOLAT, 1., 20., 80., 'BAD LATITUDE --')
CALL GETSSX(OEOLON, 1., 60., 100., 'BAD LONGITUDE --')
CALL GETSIN(ITEMP, +1.-1., 'EXTRA ALIGN SPECIFICATION --')
IF(NOSAVE.NE.NOTOTL) GO TO 900
CALL ALIGN

C
C CONFIRM GEOGRAPHIC COORDINATES
C
WRITE(6,445) MSALIN, MSASAM, OEOLAT, OEOLON
445 FORMAT(' ALIGN. SCAN. ',15.,' ',15.,
& ' DEGREES. ',F9.5, ' LAT. ',F9.5, ' LON')
GO TO 900

C
C GET/CHECK UTM COORDINATES
C
600 CF=1.E+3 3 CONVERT FROM KILOMETRES TO METRES
GO TO 620
610 CF=1. 4 METRES -- NO CONVERSION NEEDED
620 IF((UTMCMO.EQ.0.) OR.(ABS(UTMCMO).GT.180.) CALL MDWARN(
& 'UTM ZONE NOT DEFINED')
CALL GETSRL(UTME, CF.0., 1.E+6., 'BAD EASTING --')
CALL GETSRL(UTMN, CF.0., 9E+6., 'BAD NORTING --')
CALL GETSIN(ITEMP, +1.-1., 'EXTRA ALIGN SPECIFICATION --')
CALL OETSRL(UTME, CF.0., 9E+6, 'BAD COORDINATE SYSTEM --')
CALL OETSRL(UTMN, CF.0., 9E+6, 'BAD COORDINATE SYSTEM --')
CALL OETSRL(UTMCMO, CF.0., 9E+6, 'BAD COORDINATE SYSTEM --')
IF(NOSAVE.NE.NOTOTL) GO TO 900
CALL ALIGN

C
C CONFIRM UTM COORDINATES
C
WRITE(6,645) MSALIN, MSASAM, UTME, UTMN
645 FORMAT(' ALIGN. SCAN. ',15.,' ',15.,
& ' KM. ',-3P.F7.3, ' EAST. ',F7.3, ' NORTH')

C
C NORMAL RETURN
C
900 KOMO=' '
RETURN

C
C

M-6
INTERNAL SUBROUTINE ALIGN

C COMPUTE DELTAS IN STM COORDINATES BETWEEN OLD ALIGNMENT & NEW ALIGNMENT
C
IF(ABS(STMCHD-GEOLON).GT.4.) STMCHD=GEOLON
CALL U40(STME,STHN, GEDLAT,GEOLON,STMCHD)
DSTM=STME-STMN4C(CORLIN,CORSAM)
DSTHN=STHN-STHN4C(CORLIN,CORSAM)

C USE STM DELTAS TO MODIFY TRANSFORMATION COEFFICIENTS
C
CORSTM(3)=CORSTM(3)+DSTMN
CORSTM(8)=CORSTM(8)+DSTME
CALL REVERT(CORSTM,STMCH)

C MARK ORIGIN AS DESTROYED
C
KSY0NN(NORIO)=0 & ORIGINS IN DIFFERENT SYSTEMS NO LONGER CONSISTENT
CALL MONOTE(ORIGIN MUST BE RE-ENTERED)

C RECOMPUTE SCENE CENTER
C
CALL 04A(CTRLAT,CTRLON,CTRLIN,CTRSAM)

C RETURN
END
SUBROUTINE KMOATTI: 0 GET/CHECK PLATFORM ATTITUDE
U KOM01: 0: FIRST 3 CHAR OF COMMAND O: SPACES

HISTORY

E H SCHLOSSER LEC 08/05/73 NUMERIC OPTION
E H SCHLOSSER LEC 12/07/75 ALPHANUMERIC COMMAND
E H SCHLOSSER LEC 07/17/79 DELETE RETURN K
E H SCHLOSSER LEC 02/20/79 REVISE DOCUMENTATION & GETS.. CALLS
E H SCHLOSSER LEC 11/07/79 ROUND PITCH & ROLL TO 2 PLACES

METHOD

UPDATE ATTITUDE (PITCH & ROLL) FROM UNIT S, IF SPECIFIED, AND CONFIRM.

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

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

UNGETS 0 BACK UP ONE DATA FIELD ON UNIT S
GETSKH 0 GET CHARACTER STRING DATA FIELD FROM UNIT S
GETSRL 0 GET REAL DATA FIELD FROM UNIT S
GETSIN 0 GET INTEGER DATA FIELD FROM UNIT S
MOWARN 0 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
SPLIT 0 SPLIT REAL INTO SIGN, INTEGER, DECIMAL

EXCEPTIONS

1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   -2.0 <= PITCH <= +2.0
   -2.0 <= ROLL <= +2.0

2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS
INCLUDE KOMXOT.LIST  COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST  COMMON CRTS SCENE PARAMETERS
INCLUDE NULCST.LIST  DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

INTEGER ITEMP  TEMPORARY
INTEGER ISPITD  SIGN OF PITCH (+' OR '-' )
INTEGER INPITD  INTEGER PART OF PITCH
REAL APITD  DECIMAL PART OF PITCH
INTEGER ISROL0  SIGN OF ROLL (+' OR '-' )
INTEGER INROL0  INTEGER PART OF ROLL
REAL AROL0  DECIMAL PART OF ROLL

PROCEDURE

CALL TRACE

GET/CHECK PITCH & ROLL

CALL GETSR(PITDEG. 1.-2..2..2.. BAD PITCH ---)
IF(PITDEG.LE.0.00 TO 120
   CALL UNGETS
   CALL GETSKH(ITEMP.(1). NULCST)
   IF(ITEMP.NE.'-') CALL MOWARN( 'PITCH UNSIGNED. + ASSUMED')
120 CALL GETSR(ROLDEG. 1.-2..2..2.. BAD ROLL --)
   IF(ROLDEG.LE.0.00 TO 150
   CALL UNGETS
   CALL GETSKH(ITEMP.(1). NULCST)
   IF(ITEMP.NE.'-') CALL MOWARN( 'ROLL UNSIGNED. + ASSUMED')
150 CALL GETSRN(ITEMP. +1.-1. 'EXTRA ATTITUDE SPECIFICATION ---')

CONFIRM PITCH & ROLL

PITDEG=0.01*INT(100.*PITDEG+SIGN(0.5,PITDEG))  ROUND TO 2 DECIMAL PLACES
CALL SPLIT(PITDEG.ISPITD.INPITD.APITD)
ROLDEG=0.01*INT(100.*ROLDEG+SIGN(0.5,ROLDEG))  ROUND TO 2 DECIMAL PLACES
CALL SPLIT(ROLDEG.ISROL0.INROL0.AROL0)
IF(NCONFIRM.NE.0) WRITE(6,165)
6 ISPITD.INPITD.APITD.ISROL0.INROL0.AROL0
165 FORMAT( ' ATTITUDE. '..A11.F3.2. ' PITCH. '..A11.F3.2. ' ROLL')

END
SUBROUTINE KNOCEN: A GET/CHECK SCAN COORDINATES OF SCENE CENTER
U KONG: 0: 8: FIRST 3 CHAR OF COMMAND 8: SPACES

HISTORY
--------
E N SCHLOSSER  LEC  12/14/79  RQMTS/DESIGN/DOC

METHOD
-------
UPDATE SCENE CENTER FROM UNIT 5. IF SPECIFIED. AND CONFIRM.

MACHINE-DEPENDENT CODE
------------------------
NONE.

EXTERNAL REFERENCES
---------------------
GETSKH  & GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN  & GET INTEGER DATA FIELD FROM UNIT 5
GETSRL  & GET REAL DATA FIELD FROM UNIT 5
WARNS   & GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
HDEHARN & PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE

EXCEPTIONS
----------
1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
SCANN SPECIFICATION MISSING OR MISSPELLED

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   1000. < SCAN LINE < 2000.
   1000. < SCAN SAMPLE < 2000.

3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS
---------------------
INCLUDE KWXOT.LIST  & COMMON PROGRAM EXECUTION SWITCHES. COUNTERS
INCLUDE KEKNER.LIST  & COMMON ERTS SCENE PARAMETERS
INCLUDE NULCST.LIST  & DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS
OAN PACKAGE APPENDIX M
COMMAND ROUTINES

C

C INTEGER NOSAVE
INTEGER KTEMP

C PROCEDURE

C

C CALL TRACE

C GET/CHECK SCAN (OSFC-ADJUSTED) COORDINATES

C NOSAVE=NOTOTL
KTEMP=' NUL'
CALL GETSKH(KTEMP,(3), NULCST)
IF(KTEMP.EQ.' NUL') GO TO 400
NO SPECS -- CONFIRM
IF(KTEMP.NE.'SCA') CALL WARN( 'COORDINATE SYSTEM NOT SCAN --')
IF(KTEMP.NE.'SCA') GO TO 400
CALL GETSRL(CTRIN, 1..1000..2000..'BAD LINE --')
CALL GETSRL(CTRSAM, 1..1000..2000..'BAD SAMPLE --')
CALL GETSIN(KTEMP, +1,-1..'EXTRA CENTER SPECIFICATION --')

C CONFIRM CENTER COORDINATES

C 400 IF(MCFIRM.NE.0) WRITE(6,449) CTRIN,CTRSAM
449 FORMAT(' CENTER, SCAN. ',FG.1,'..',FG.1)

C NORMAL RETURN

C 900 KOMD=' RETURN
END
SUBROUTINE KNOCHA: A GET/CHECK RAW/TRANSFORMED CHANNEL NUMBER(S)
U KNOBA) 0 1: FIRST 3 CHAR OF COMMAND 0: SPACES

HISTORY

E H SCHLOSSER LEC 08/27/73 NUMERIC OPTION
E H SCHLOSSER LEC 12/09/79 ALPHANUMERIC COMMAND
E H SCHLOSSER LEC 07/17/79 DELETE RETURN
E H SCHLOSSER LEC 02/11/79 REVISE GETS, CALLS
J C CRISP LEC 11/29/79 UPGRADE DOCUMENTATION
J C CRISP LEMSCO 02/19/80 DON'T ASSIGN TEMP BUFFER POINTERS

METHOD

UPDATE TRANSFORMATION TYPE & CHANNELS FROM UNIT S. IF SPECIFIED.
AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSKH 0 GET CHARACTER STRING DATA FIELD FROM UNIT S
UNGETS 0 BACK UP ONE DATA FIELD ON UNIT S
GETSIN 0 GET INTEGER DATA FIELD FROM UNIT S
WARS 0 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT S

EXCEPTIONS

1. IF TRANSFORMATION TYPE IS NOT SPECIFIED, THEN 'RAW' IS ASSUMED.

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
"WARNING" DIAGNOSTICS:
   TRANSFORMATION TYPE = 'RAW'/'LIN'/'POL'
   1 <= RAW CHANNEL <= NCHNA
   1 <= TRANSFORMED CHANNEL <= 2

3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KONQUT.LIST 0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
DAM PACKAGE APPENDIX H
COMMAND ROUTINES

- INCLUDE KONNER.LIST 0 COMMON ERTS SCENE PARAMETERS
- INCLUDE KONKL.LIST 0 COMMON CLASSIFICATION INFO
- INCLUDE KONKIR.LIST 0 COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
- INCLUDE NULCST.LIST 0 DEFINE NUL CHARACTER STRING

C
C LOCAL DECLARATIONS
C

INTEGER KTYPE 0 TRANSFORMATION TYPE SPEC OR CONFIRMATION
INTEGER NDSAVE 0 TEMPORARY SAVE FOR CONTENTS OF NDTOTL ON ENTRY
INTEGER NLMAX 0 MAXIMUM NUMBER OF LIMIT CHANNELS
INTEGER NSF 0 BUFFER NUMBER
INTEGER INTTEMP 0 TEMPORARY

C
C PROCEDURE
C
C CALL TRACE

C ANY SPECIFICATIONS FROM USER?

KTYPE=' NUL'
CALL GETSKH(KTYPE,11, NULCST)
IF(KTYPE.EQ.' NUL') GO TO 600 0 NO SPECS. SO CONFIRM CURRENT ONES
CALL UNGETS

C SAVE DIAGNOSTIC COUNT. DELETE OLD TRANSFORM TYPE. CHANNELS & RADIANCE LIMITS

NDSAVE=NDTOTL
IRTYP='NUL'
NLINCH=0
DO 120 I=1,9
LINCH(INI)=999 0 NULL FLAG
NFBCH(INI)=999
LCOLO(INI)=9999
LCOVC(INI)=9999
120 CONTINUE

C GET CHANNEL TRANSFORMATION TYPE

CALL GETSKH(IRTYP,13, NULCST)
IF(IRTYP.EQ.' NUL') GO TO 220 0 POSTITIVE WORD STARTS WITH ALPHA CHARACTER

C NO CHANNEL TYPE -- ASSUME 'RAM' (UNTRANSFORMED)

IRTYP='RAM'
CALL UNGETS

C CHECK CHANNEL TYPE & SET MAX ALLOWABLE NUMBER OF EXPLICIT CHANNELS

N-13
C

220 NLCHAX=0  0 MAXIMUM NUMBER OF TRANSFORMED CHANNELS
IF(IRITYP.EQ. 'LIN') GO TO 300  0 LINEAR TRANSFORMATION
IF(IRITYP.EQ. 'POL') GO TO 300  0 POLAR TRANSFORMATION
NLCHAX=NLCHAX  0 MAXIMUM NUMBER OF RAW CHANNELS
IF(IRITYP.EQ. 'RAW') GO TO 300  0 UNTRANSFORMED
CALL WARNS( 'BAD CHANNEL TYPE --')

C

C GET/CHECK EXPLICIT CHANNEL NUMBERS
C

300 GO 380 HD=1,NLCHAX
CALL GETSIN(INCH(NDF), 1,NLCHAX, 'BAD CHANNEL --')
IF(INCH(NDF).EQ.999) GO TO 360  0 NO CHANNEL OR BAD CHANNEL
NLINCH=NLINCH+1
IF(NDF.EQ.1) GO TO 380
DO 340 N=2,NDF
IF(INCH(N-1).EQ.INCH(NDF)) CALL WARNS( 'DUPLICATE CHANNEL --')
340 CONTINUE
380 CONTINUE
CALL GETSIN(INCH TEMP. +1,-1, 'TOO MANY CHANNELS --')

C

C CHECK FOR DIAGNOSTICS

C

IF(INOTL.EQ.NDSAVE) GO TO 500
IRITYP='NUL'
NLINCH=0
INCH(1)=999
NDFCH(1)=999
GO TO 500

C

C CONFIRM CHANNEL TYPE AND CHANNEL NUMBERS
C

500 IF(INCHFRM.EQ.0) GO TO 500
IF(IRITYP.NE.'LIN') GO TO 620
KTYPE='LINEAR'
GO TO 650
620 IF(IRITYP.NE.'POL') GO TO 630
KTYPE='POLAR'
GO TO 650
630 KTYPE='RAW'

C

C WRITE(6,695) KTYPE,(INCH(N),N=1,NLINCH)
695 FORMAT( CHANNEL, '., AB., ..., 12, 1S, 12,
& T23., ..., T25.12,
& T27., ..., T33.12,
& T31., ..., T37.12,
& T35., ...)
DAM PACKAGE APPENDIX M
COMMAND ROUTINES

RETURN

END
SUBROUTINE DETCHA( & GET/CHECK DETECTION CHANNEL NUMBER(S))

U KOND1 & 1: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY

E H SCHLOSSER  LEC  08/27/73  NUMERIC OPTION
E H SCHLOSSER  LEC  12/09/75  ALPHANUMERIC COMMAND
E H SCHLOSSER  LEC  07/17/78  DELETE RETURN K
E H SCHLOSSER  LEC  02/11/79  REVISE GETS.. CALLS
J C CRISP      LEC  11/29/79  UPGRADE DOCUMENTATION
J C CRISP      LEMSCO 02/19/80  DON'T ASSIGN TEMP BUFFER POINTERS
J C CRISP      LEMSCO 05/24/80  SPLIT INTO KHDDCH & DETCHA

METHOD

UPDATE DETECTION CHANNEL NUMBER(S) FROM UNIT 5, IF SPECIFIED.
AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

- GETSKH   & GET CHARACTER STRING DATA FIELD FROM UNIT 5
- UNGETS   & BACK UP ONE DATA FIELD ON UNIT 5
- GETSIN   & GET INTEGER DATA FIELD FROM UNIT 5
- WARNS    & GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
- LOGICAL TRUAL & TRUE IF CHARS IN CST ARE ALPHABETIC

EXCEPTIONS

1. IF CHANNEL TYPE IS NOT SPECIFIED, THEN 'DET' IS ASSUMED.

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   "WARNING" DIAGNOSTICS:
   - TRANSFORMATION TYPE = 'DET'
   - 1 <= CHANNEL NUMBER <= NCHERCH

3. AN EXTRA SPECIFICATION GENERATES A "WARNING" DIAGNOSTIC.

GLOBAL DECLARATIONS
DAM PACKAGE APPENDIX N
COMMAND ROUTINES

INCLUDE KOMXOT.LIST
INCLUDE KOMNER.LIST
INCLUDE KOMKLS.LIST
INCLUDE NULCST.LIST

LOCAL DECLARATIONS

INTEGER KHTYPE
INTEGER NOSAVE
INTEGER NBF
INTEGER INTEMP

PROCEDURE

CALL TRACE

ANY SPECIFICATIONS FROM USER?

KHTYPE=' NUL'
CALL GETSKH(KHTYPE,(1), NULCST)
IF(KHTYPE.EQ.' NUL') GOTO 600 NO SPECS. SO CONFIRM CURRENT ONES
CALL UNGETS

SAVE DIAGNOSTIC COUNT. DELETE OLD CHANNEL NUMBERS & RADIANCE LIMITS

NOSAVE=NDTOTL
NOSAVE=NOSAVE
DO 120 N=1,5
   L='CHIN'=.999 NULL FLAG
   NBFCHR(N)=999
   LCYLO(N)=999
   LCYHI(N)=999
120 CONTINUE

GET/CHECK CHANNEL TYPE -- IF NONE, ASSUME 'DET'

CALL GETSKH(KHTYPE,(3), NULCST)
IF(ISTRUE(KHTYPE,1)) GOTO 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 WARN5(
   'CHANNEL TYPE NOT DETECTION --')

GET/CHECK EXPLICIT CHANNEL NUMBERS

DO 360 NBF=1, NERCH
   CALL GETSIN(LIMCH(NBF), NERCH, 'BAD CHANNEL --')
   IF(LIMCH(NBF).EQ.999) GOTO 360 NO CHANNEL OR BAD CHANNEL

M-17
CAN

►

ACKAOE APPENDIX M

OETCHA

COMMAND ROUTINES

NLINCH=NLINCH+1
IF(NBF.EQ.1) GO TO 380
GO 340 N=2,NBF

IF(LINCH(N-1).EQ.LINCH(NBF)) CALL WARN( 'DUPLICATE CHANNEL --')

340 CONTINUE
360 CONTINUE
CALL GETSN(INTERN, '1,-1,'TOO MANY CHANNELS --')

C

C CHECK FOR DIAGNOSTICS
C
IF(NOTOTL.EQ.NDSAVE) GO TO 600
NLINCH=0
LINCH(1)=999
NBFCHR(1)=999
GO TO 900

C

C CONFIRM CHANNEL TYPE AND CHANNEL NUMBERS
C
600 IF(MCFIRM.EQ.0) GO TO 900
WRITE(6,655) (LINCH(N),N=1,NLIMCH)

655 FORMAT(' CHANNEL, DETECT. '12.25.12.
 & T27. '' T33.12. 
 & T31. '' T37.12. 
 & T35. '')

C

C NORMAL RETURN
C
900 KOMD=''
RETURN
END
SUBROUTINE KMOCLE  A CLEAR WARNINGS/ERRORS
U KMND)  8 1: FIRST 3 CHAR OF COMMAND 0: SPACES

METHOD

CLEAR COUNT OF WARNINGS OR OF FATAL ERRORS.

EXTERNAL REFERENCES

GETSKN  A GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSN  A GET INTEGER DATA FIELD FROM UNIT 5
WARNS  A GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
MDCLRW  A CLEAR WARNING(S)
MDCLRF  A CLEAR FATAL ERROR(S)

EXCEPTIONS

1. A MISSING SPECIFICATION IS INTERPRETED AS IF WARNING HAD BEEN SPECIFIED
2. AN INVALID OR EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC MESSAGE.

GLOBAL DECLARATIONS

INCLUDE KOMQIT.LIST  A COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST  A DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

INTEGER ITEMP  A TEMPORARY
PROCEDURE
---------

CALL TRACE

GET/CHECK SPECIFICATION

ITEMP='WAR'
CALL GETSKMITEMP(3), 'NO CLEAR SPEC -- WARNING ASSUMED')
IFITEMP.EQ.'WAR' GO TO 200
IFITEMP.EQ.'ERR' GO TO 300
     CALL WARN('BAD CLEAR SPECIFICATION --')
     GO TO 900

CLEAR WARNINGS

200 CALL MDECLR(NULCST)
     IF(MCFIRM.NE.0) WRITE(6,215)
     215 FORMAT(' CLEAR, WARNINGS')
     GO TO 900

CLEAR FATAL ERRORS

300 CALL MDECLF(NULCST)
     IF(MCFIRM.NE.0) WRITE(6,315)
     315 FORMAT(' CLEAR, ERRORS')

COMMON RETURN

900 CALL GETSNITEMP, 'I, -1, 'EXTRA CLEAR SPECIFICATION --')
     KOMON
     RETURN
     END
SUBROUTINE KMOCOL  8 GET/CHECK COLOR(S)
U KOM01  8 1: FIRST 3 CHAR OF COMMAND  0: SPACES

HISTORY
-------
E H SCHLOSSER  LEC  05/26/79  MODIFY FROM KHDOSYN

METHOD
-------
GET. CHECK. INTERPOLATE IF NECESSARY, AND STORE THE
INTEGER-COLOR-EQUIVALENTS (I-K-E) FOR CRT PICTURES IN CHARACTER 6
OF THE KSYM ARRAY ELEMENTS.

MACHINE-DEPENDENT CODE
------------------------
DIMENSION SPECIFICATIONS ASSUME 6 CHARACTERS PER INTEGER.

EXTERNAL REFERENCES
------------------
GETSKH  8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN  8 GET INTEGER DATA FIELD FROM UNIT 5
WARNS  8 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
UNGETS  8 BACK UP ONE DATA FIELD ON UNIT 5
MOWARN  8 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
MOVCST  8 MOVE CHARACTER STRING
14KOLR  8 INTEGER-COLOR-EQUIVALENT FOR COLOR
KOLR41  8 COLOR FOR INTEGER-COLOR-EQUIVALENT
PUTICE  8 PUT INTEGER-CHARACTER-EQUIVALENT INTO CHARACTER STRING
GETICE  8 GET INTEGER-CHARACTER-EQUIVALENT FROM CHARACTER STRING

EXCEPTIONS
----------
1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
COLOR MODE SWITCH (MCOLOR IN KOMXQT) NOT ON
INVALID COLOR (SEE 14KOLR & KOLR41)

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
'WARNING' DIAGNOSTICS:
0 <= NUMBER <= ISYMHI (SEE KOMSYM-PROC)

GLOBAL DECLARATIONS
---------------------
INCLUDE KOMXQT.LIST  8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
LOCAL DECLARATIONS


INTEGER NOSAVE
INTEGER MINKLR(2), MAXKLRL(2) & "MINIMUM", "MAXIMUM" COLOR (12 CHARACTERS)
INTEGER MINIKE, MAXIKE & "MINIMUM", "MAXIMUM" INTEGER-COLOR-EQUIVALENTS
REAL AMIINIK & A FLOOR OR CEILING OF MINIKE FOR INTERPOLATING UP OR DOWN
INTEGER INCIKE & A INCREMENT DIRECTION MINIKE TO MAXIKE (+1 OR -1)
INTEGER NUM & A NUMBER (RAD/DENSITY/COUNT/CLASS, ETC.) FOR COLOR
INTEGER MINNUM, MAXNUM & "MINIMUM", "MAXIMUM" NUMBER
INTEGER INCNUM & A INCREMENT DIRECTION MINNUM TO MAXNUM (+1 OR -1)
INTEGER INTEMP & A TEMPORARY
REAL COINUM & A COLORS PER NUMBER (FOR INTERPOLATION)

PROCEDURE
---------

CALL TRACE

INITIALIZE

IF(MCOLOR.NE.0) GO TO 120
CALL MDWARN( "COLOR COMMAND NOT ALLOWED (COLOR MODE NOT ON)"
GO TO 900
120 NOSAVE=NOTOTL

GET OPTIONAL "MINIMUM" COLOR FROM 1ST SPEC FIELD

MINKLRL(1)="NUL"
CALL GETSKM(MINKLR(12), NULCST)
IF(MINKLR(1).EQ."NUL") GO TO 850 & NO SPECS
MINIKE=MAXIKE
CALL 1NKLR(MINIKE, MINKLR) & CONVERT COLOR TO I-K-E
IF(MINIKE.EQ.MAXIKE) CALL WARN( "BAD FIRST COLOR --")

GET REQUIRED "MINIMUM" NUMBER (INTEGER) FROM 2ND SPEC FIELD

MINNUM-MAXNUM
CALL GETSIN(MINNUM, "SYMNI," "BAD FIRST NUMBER --")
DAN PACKAGE APPENDIX M
COMAND ROUTINES

C INITIALIZE IMPLICIT 'MAXIMUM' COLOR AND NUMBER
CALL MOVCST(MAXKLR(1), (12), MINKLR(1), (12), ' ')
MAXNUM=MINNUM

C IS 3RD SPEC FIELD INTEGER?

C INTTEMP-MAXINT
CALL GENST(INTTEMP, -9999., 9999., NULCST)  & STILL MAXINT IF NOT INTEGER
CALL UNGETS

C IF 3RD SPEC FIELD IS NOT INTEGER, THEN GET OPTIONAL 'MAXIMUM' COLOR FROM IT

C IF(INTTEMP.EQ.MAXINT) CALL GENST(MAXKLR(12), NULCST)
MAXIKE-MAXINT
CALL UNOLR(MAXIKE, MAXKLR) & CONVERT COLOR TO 1-K-E
IF(MAXIKE.EQ.MAXINT) CALL WARN('BAD SECOND COLOR --')

C GET OPTIONAL 'MAXIMUM' NUMBER (INTEGER) FROM NEXT SPEC FIELD

C CALL GENST(MAXNUM, 0., ISYMHI.'BAD SECOND NUMBER --')

C COMPUTE COLORS PER NUMBER

C INCIKE+1
IF(MINIKE.GT.MAXIKE) INCIKE=-1
MAXNUM=MIN(MAXNUM, ISYMHI)
INCNUM+1
IF(MINNUM.GT.MAXNUM) INCNUM=-1
C0INUM=FLOAT(MAXIKE-MINIKE+INCIDE) & PREPARE TO INTERPOLATE ...
& FLOAT(MAXNUM-MINNUM+INCNUM) & ... OVER INCLUSIVE RANGE
IF(ABS(C0INUM).GT.1.) CALL MOWARN('MORE THAN 1 COLOR PER NUMBER')

C CHECK FOR EXTRA SPEC FIELDS & FOR DIAGNOSTICS

C CALL GENST(INNNUM, +1., -1., 'EXTRA COLOR SPECIFICATION --')
IF(NOSAVE.NE.NOTOTL) GO TO 900

C INTERPOLATE 1-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 ON FROM HIGH SIDE OF MINIKE
DO 300 NUM=MINNUM.MAXNUM.INCNUM
CALL PUTICE(KSYM(NUM+1), (6), & INTERPOLATE 1-K-E
C IFIX(AMINIK+FLOAT(NUM-MINNUM)+COINUM)
300 CONTINUE
CONFIRM COLORS

IF(INCFIRM.EQ.0) GO TO 900
   CALL KOLR1(MINLR, MINKE) & CONVERT 1-K-E TO COLOR
   CALL KOLR1(MAXLR, MAXKE) & CONVERT 1-K-E TO COLOR
   WRITE(8,325) MINLR,MINNUM,MAXLR,MAXNUM
325   FORMAT( COLOR, '.ZAG.', '.J3.', '.ZAG.', '.J3.')
   GO TO 900

NO COLORS SPECIFIED -- CONFIRM ALL COLORS

900 IF(INCFIRM.NE.0) CALL ALLKLR

RETURN

INTERNAL SUBROUTINE ALLKLR & CONFIRM ALL CURRENT COLORS

INTEGER NXTIKE & NEXT 1-K-E

MINNUM=0
   CALL GETICE(MINKE, KSYM(MINNUM))
   DO 200 MAXNUM=0,15MINK
   CALL GETICE(NXTIKE, KSYM(MAXNUM+1))
   IF(NXTIKE.EQ.MINIKE) GO TO 200 & NEXT 1-K-E SAME AS CURRENT
   CALL KOLR1(MINLR, MINIKE) & CONVERT 1-K-E TO COLOR
   IF(MINNUM.EQ.MAXNUM)
     WRITE(8,145) MINLR,MINNUM
145   FORMAT( COLOR, '.ZAG.', '.J3.')
   IF(MINNUM.NE.MAXNUM)
     WRITE(8,185) MINLR,MINNUM,MAXNUM
185   FORMAT( COLOR, '.ZAG.', '.J3.', '.J3.')
   MINNUM=MAXNUM+1
   MINIKE=NXTIKE
200 CONTINUE

RETURN

END
SUBROUTINE KHDSCP: 0 GET/CHECK NUMBER OF OUTPUT COPIES
U KHDSCP: 0 I: FIRST 3 CHAR OF COMMAND 0: SPACES

C HISTORY

C

C E N SCHLOSSER LEC 12/09/79 ORIGINAL CODE
C E N SCHLOSSER LEC 07/18/79 DELETE RETURN K
C E N SCHLOSSER LEC 02/20/79 REVISE GETS.. CALLS
C E N SCHLOSSER LEC 11/10/79 UPGRADE DOCUMENTATION

C METHOD

C

UPDATE NUMBER OF OUTPUT COPIES FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

C MACHINE-DEPENDENT CODE

C

NONE.

C EXTERNAL REFERENCES

C

GETSIN 0 GET INTEGER DATA FIELD FROM UNIT 5

C EXCEPTIONS

C

1. IF THE NUMBER OF COPIES SPECIFIED IS < 0 OR > 5, A 'WARNING'
   DIAGNOSTIC IS ISSUED AND THE CURRENT NUMBER OF COPIES IS NOT
   CHANGED.

2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC MESSAGE.

C GLOBAL DECLARATIONS

C

INCLUDE KOMXOT.LIST 0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMALT.LIST 0 COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS

C LOCAL DECLARATIONS

C

INTEGER ITMP 0 TEMPORARY

C

C

C

M-29
CALL TRACE

CALL GETSIN(NCOPY, 0,9,'BAD NUMBER OF COPIES (MORE THAN 9) --')
CALL GETSINITEMP, +1,-1,'EXTRA COPY SPECIFICATION --')
IF(NCFIRM,HE,0) WRITE(8,129) NCOPY
129 FORMAT(' COPIES, ',I2)
KOND=''
RETURN
END
SUBROUTINE KMOCOU: 8 GET/CHECK COUNT PER PIXEL
U KOMD) 0 1: FIRST 3 CHAR OF COMMAND 0 1: SPACES

HISTORY

E M SCHLOSSER  LEC  07/17/78  IN DENSITY COMMAND (KOMDEN)
E M SCHLOSSER  LEC  03/22/78  MAKE SEPARATE COMMAND

METHOD

UPDATE COUNT PER PIXEL FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSIN  8 GET INTEGER DATA FIELD FROM UNIT 5

EXCEPTIONS

1. COMMAND SPECIFICATIONS ARE OPTIONAL, BUT IF PRESENT ARE CHECKED FOR
   VALIDITY AND RESULT IN WARNING DIAGNOSTIC IF BAD.

2. KOMD IS NOT CHECKED FOR VALIDITY.

GLOBAL DECLARATIONS

INCLUDE KONKOT.LIST  8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONKLS.LIST  8 COMMON CLASSIFICATION INFO

LOCAL DECLARATIONS

INTEGER INTEMP  8 TEMPORARY

PROCEDURE
CALL TRACE

C

GET/CHECK COUNT SPECIFICATION

CALL GETSN(KTIPIX, 0, 2, 'BAD COUNT PER PIXEL --')
CALL GETSI(INTEP, -1, -1, 'EXTRA COUNT SPECIFICATION --')
IF(INFIRM.NE.0) WRITE(0,129) KTIPIX
129 FORMAT('COUNT, 12.2 PER PIXEL')
KOND=''
RETURN
END
SUBROUTINE KMOCROI	 & CROSSTAB FREQUENCY OF DATA IN PREVIOUS WINDOW
U KOMDI	 & I: FIRST 3 CHARS OF COMMAND O: SPACES


HISTORY
-------
E H SCHLOSSER	 LEC	 05/20/74	 ORIGINAZ CODE IN DISTAB
E H SCHLOSSER	 LEMSCO	 05/28/80	 ADAPT INTO KMOCRO

METHOD
-------
PRINT CROSSTABULATION OF INTENSITY BY COLOR FROM FREQUENCIES IN KOMTBL.

MACHINE-DEPENDENT CODE
-----------------------------
ASSUMES 12 CHARACTERS PER DOUBLE PRECISION VARIABLE.

EXTERNAL REFERENCES
---------------------
MDNOTE	 & PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
MDCLRN	 & CLEAR 'WARNING' DIAGNOSTICS
IDERT	 & IDENTIFY ERTS SCENE
KOLR41	 & COLOR NAME (STRING) FOR INTEGER-COLOR-EQUIVALENT

EXCEPTIONS
-----------
1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   COLOR MODE SWITCH (MCOLOR IN KOMQX) NOT ON
2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS
----------------------
INCLUDE KOMQXT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST & COMMON ERTS PARAMETERS
INCLUDE KOMKL5.LIST & COMMON CLASSIFICATION INFO
INCLUDE KOMTBL.LIST & COMMON MULTI-PURPOSE TABLE
INCLUDE NULCST.LIST & DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS
---------------------
INTEGRITEMP & TEMPORARY
DOouble Precision Kolman 8 Color Name (12 CharS)
Integer KFrSUB(10) 8 Freq SubTotals

C C Procedure
C
C CALL TRACE
C
C Check if KOMTBL 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.01) OR. 2 NO CLASS TYPE FROM PREVIOUS DISPLAY
& (KTBLTY.NE.'FREQ') 2 NO FREQ TABLES LOADED FROM PREVIOUS DISPLAY
& CALL MDWARN:  'NO PREVIOUS WINDOW TO CROSSTAB'
C
C Drain Specs for Current Command
C
CALL GETSIN(INTEMP. *1.-1.'EXTRA CROSSTAB SPECIFICATION --')
C
C Check for Diagnostics
C
IF(MDA TC.NE.0) GO TO 900  B DATA/CHECKOUT Mode
IF(NOTOTL.NE.0) GO TO 800
C
C Use Previous Window Number for Page Heading
C
NWTEMP=NWNDOW  B Save Current Window Number
NWNDOW=KTBLNW  B Window Number From Frequency Tables
C
C Print Page/Window Headings
C
WRITE(6,115) NWNDOW,MTERAL
115 FORMAT(' WINDOW NUMBER ',J3,6X,'CROSSTAB',6X,4A6)
CALL MDUNIT(4,10)
WRITE(10,115) NWNDOW,MTERAL
C
C Restore Current Window Number
C
NWNDOW=NWTEMP
C
C Print Table Headings
C
CALL IDERT(6)
CALL IDERT(10)
IF(MBATCH.EQ.0) WRITE(6,125) WRITE(10,125)
125 FORMAT(*'COLOR INTENSITY'/
  & 80% 70% 60% 50% 40% 30% 20% 10% TOTAL'
& WRITE(6,135) WRITE(10,135)
135 FORMAT(1X)

C
C INITIALIZE SUBTOTALS
C DO 340 IIE=0.9 KFRSUB(IIE+1)=0
340 CONTINUE

C C PRINT CROSSTABULATIONS
C DO 600 IIE=0.10 8 BLUE THROUGH MAGENTA
CALL KOLR41(KOLNAM. IIE)
KFRTOT=0
DO 400 IIE=0.9 8 10% THROUGH 100%
KFRTOT=KFRTOT+KFRSUB(IIE+1)
KFRSUB(IIE+1)=KFRSUB(IIE+1)+KFRSUB(IIE+1)
400 CONTINUE
IF(MBATCH.EQ.0) WRITE(6,425)
& KOLNAM. (KFRSUB(IIE+1)+1). IIE=0.91 KFRTOT
WRITE(10,425) KOLNAM. (KFRSUB(IIE+1)+1). IIE=0.91 KFRTOT
425 FORMAT(1X.9.1X.1016.17)
600 CONTINUE

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)
& WRITE(10,615) (KFRSUB(IIE+1)+1). IIE=0.91 KFRTOT
615 FORMAT('TOTAL'.5X.1016.17)
GO TO 900

C C CHECK DIAGNOSTIC COUNTERS
C 800 IF(NOWARN.EQ.0) GO TO 820 CALL NDNOTE( 'PREVIOUS WARNINGS -- NO CROSSTAB GENERATED') IF(MBATCH.EQ.0) WRITE(6,815)
815 FORMAT('...TRY AGAIN')
GO TO 890
820 IF(NOFAIL.EQ.0) GO TO 850

M-31
CALL MONOTE('PREVIOUS FATAL ERRORS -- NO CROSSTAB GENERATED')
00 TO 890
890 IF(MCHECK.EQ.0) GO TO 890
CALL MONOTE('CHECKOUT MODE -- NO CROSSTAB GENERATED')

C CLEAR WARNINGS
C
890 CALL MOCLEAN(NULCST)
C
RETURN
C
900 KOMO=''
RETURN
END
SUBROUTINE KMODEN: GET/CHECK DENSITY LIMITS

U KMOD: 8 I: FIRST 3 CHAR OF COMMAND 0: SPACES

HISTORY

E C H SCHLOSSER	 LEC	 07/03/73	 NUMERIC OPTION
E C H SCHLOSSER	 LEC	 11/29/75	 ALPHANUMERIC COMMAND
E C H SCHLOSSER	 LEC	 07/17/78	 ADD COUNT SPECIFICATION
E C H SCHLOSSER	 LEC	 03/22/79	 CONFIRM COUNT AS SEPARATE COMMAND
E C H SCHLOSSER	 LEC	 11/13/79	 DOCUMENT SPECIFICATION RANGES

METHOD

UPDATE MINIMUM & MAXIMUM DENSITY FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSIN 8 GET INTEGER DATA FIELD FROM UNIT 5

EXCEPTIONS

1. IF THE CURRENT DETECTION FILE TYPE IS NOT 'DEN', THEN A WARNING IS
   GENERATED.

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   0 <= MIN DENSITY <= 19
   MIN DENSITY <= MAX DENSITY <= 19
   0 <= COUNT/PIXEL <= 12

GLOBAL DECLARATIONS

INCLUDE KOMXGT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST 8 COMMON CLASSIFICATION INFO

LOCAL DECLARATIONS

H-33
SUBROUTINE APPENDIX M

INTEGER INTEMP, I, TEMPORARY

PROCEDURE
---------

CALL TRACE

C DENSITY COMMAND IS VALID ONLY IF DETECTION FILE TYPE IS 'DEN'
C IF (KLISTYP .EQ. 'DEN') GO TO 200

CALL MDHARM( 'INVALID WITHOUT DENSITY DETECTION FILE --'
GO TO 900

C GET/CHECK MINIMUM DENSITY
C 200 LCVLOI = MAX(I0, LCVLOI)
LCVLOI = MIN(I19, LCVLOI)
CALL OCISIN(LCVLOI, 0,19,'BAD MINIMUM DENSITY --'

C GET/CHECK MAXIMUM DENSITY
LCVHI1 = MAX(I0, LCVHI1)
LCVHI1 = MIN(I19, LCVHI1)
CALL OCISIN(LCVHI1, 0,19,'BAD MAXIMUM DENSITY --'

C GET/CHECK COUNT PER PIXEL
KTIPIX = MAX(I0, KTIPIX)
KTIPIX = MIN(I12, KTIPIX)
INTEMP = -9999
CALL OCISIN(INTEMP, 0,12,'BAD COUNT PER PIXEL --'
IF (INTEMP .NE. -9999) KTIPIX = INTEMP
CALL OCISIN(INTEMP, +1,-1,'EXTRA SPECIFICATION --'
IF (INTEMP .EQ. -9999) GO TO 300 & COUNT SPECIFICATION MISSING?

C CONFIRM DENSITY LIMITS & COUNT PER PIXEL
IF (MCF1RM .NE. 0) WRITE(6,225) LCVLOI, LCVHI1, KTIPIX
225 FORMAT(' DENSITY. ',I2,' MINIMUM. ',I2,' MAXIMUM'
& COUNT. ',I2,' PER PIXEL')
GO TO 900

C CONFIRM DENSITY LIMITS ONLY
C 300 IF (MCF1RM .NE. 0) WRITE(6,325) LCVLOI, LCVHI1
325 FORMAT(' DENSITY. ',I2,' MINIMUM. ',I2,' MAXIMUM')

H-34
C DONE
C
900 KOND=""
RETURN
END
SUBROUTINE KMOEXP: 6 EXPLAIN ANYTHING

U KOND) 0 1: FIRST 3 CHAR'S OF COMMAND 0: SPACES

-------------------

HISTORY

-------------------

E N SCHLOSSER LEC 12/05/75 ORIGINAL CODE
E N SCHLOSSER LEC 07/17/70 DELETE RETURN K
E N SCHLOSSER LEC 08/20/79 REVISE GETS.. CALLS
E N SCHLOSSER LEC 11/10/79 REMOVE UNIVAC FLD FUNCTION

METHOD

-------------------

FOR EACH USER SPECIFICATION:
   GET SPECIFICATION & APPEND TO 'DAM.EXP'.
   CALL RITELT TO WRITE DISK SYMBOLIC ELEMENT WITH ABOVE NAME.
   IF NO SUCH ELEMENT, TRUNCATE APPENDED SPECIFICATION TO 3 CHARACTERS
   AND CALL RITELT AGAIN.

MACHINE-DEPENDENT CODE

-------------------

DIMENSION SPECIFICATIONS ASSUME 6 CHARACTERS PER WORD.
CONDITIONAL RETURNS TO STATEMENT LABELS

EXTERNAL REFERENCES

-------------------

GETSKH 6 GET CHARACTER STRING DATA FIELD FROM UNIT 5
PUTCHR 6 PUT CHARACTER INTO CHARACTER STRING
GETCHR 6 GET CHARACTER FROM CHARACTER STRING
MOVCST 6 MOVE CHARACTER STRING
WARNS 6 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
RITADD 6 WRITE OR ADD DISK SYMBOLIC ELEMENT

EXCEPTIONS

-------------------

1. IF NO EXPLANATION FOR THE USER SPECIFICATION IS AVAILABLE, A 'WARNING'
   DIAGNOSTIC MESSAGE IS ISSUED.

GLOBAL DECLARATIONS

-------------------

INCLUDE NULCST.LIST 6 DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS
```
C INTEGERT NAMELT(5) 8 NAME OF DISK SYMBOLIC ELEMENT CONTAINING EXPLANATION
   DATA (NAMELT(N),N=1,2)" DAM.EXP-"/
   INTEGER KHR1    8 1ST CHARACTER OF USER SPECIFICATION

C PROCEDURE
C
C CALL TRACE
C
C INITIALIZE 'EXP' SPEC BEFORE TRYING TO GET 1ST SPEC
C NAMELT(3)="EXP"
C
C GET NEXT SPECIFICATION
C 120 CALL GETSKH(NAMELT(3),(17), NULCST)
   CALL PUTCHR(NAMELT(3),(18), "") 8 TERMminate WITH BLANK
   IF(NAMELT(3),EQ," NUL") GO TO 900 8 NO MORE SPECS
C
C IF 1ST CHAR IS 'I' CHANGE IT TO 'S'
C CALL GETCHR(KHR1, NAMELT(3),(11))
   IF(KHR1, EQ,'I') CALL PUTCHR(NAMELT(3),(1), "S") 8 EXEC COMMAND
C
C WRITE EXPLANATION IF AVAILABLE
C CALL RITADD(S,NAMELT,O,S180)
   GO TO 200
C
C EXPLANATION WAS NOT AVAILABLE -- TRUNCATE SPEC TO 3 CHARs & TRY AGAIN
C 180 CALL MOVCCST(NAMELT(3),(4),(13),
   "", (11),(1), "")
   CALL RITADD(S,NAMELT,O,S800)
C
C INITIALIZE NUL SPEC BEFORE TRYING TO GET NEXT SPEC
C 200 NAMELT(3)=" NUL"
   GO TO 120
C
C EXPLANATION STILL NOT AVAILABLE
C 800 CALL WARNS/ "BAD EXPLAIN SPECIFICATION --"
   GO TO 200
C
C M-37
```
C COMMON RETURN
C
900 KMOD=''
RETURN
END
SUBROUTINE KMDF1: A TERMINATE IF..FI BLOCK
U KMDF1, 8 I: FIRST 3 CHAR OF COMMAND D: SPACES

HISTORY

E H SCHLOSSER LEC 12/05/79 REQTS/DESIGN/CODE

METHOD

DO NOTHING.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSIN 3 GET INTEGER DATA FIELD FROM UNIT 9

EXCEPTIONS

1. IF ANY USER SPECIFICATION(S) ARE PRESENT, A "WARNING" DIAGNOSTIC IS GENERATED.

GLOBAL DECLARATIONS

NONE.

LOCAL DECLARATIONS

INTEGER INTEMP 8 TEMPORARY

PROCEDURE

CALL TRACE

CHECK FOR EXTRA SPECIFICATION
C  CALL GETSIN(INTEMP, '01.-1. "EXTRA F1 SPECIFICATION"')
C
C  NORMAL RETURN
C
C    KMD=""   RETURN
   END
SUBROUTINE KND0601, GET/CHECK ADJUSTED MBS GEOMETRY

UNIT 5: FIRST 3 CHAR OF COMMAND 01 SPACES

HISTORY

E H SCHLOSSER LEC 12/07/79 REQUIREMENTS
E H SCHLOSSER LEC 12/10/79 DESIGN/TEST

METHOD

UPDATE GEOMETRY FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSKN, GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN, GET INTEGER DATA FIELD FROM UNIT 5
MOWARN, PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   SCAN: SPECIFICATION MISSING OR MISSPELLED

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   0 <= SCAN LINE <= 2500
   0 <= SCAN SAMPLE <= 3500

3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KONKOT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONNER.LIST 9 COMMON ERR SCENE PARAMETERS
INCLUDE KONFIT.LIST 8 COMMON ADJUST/REGISTRATION PARAMETERS
INCLUDE NULLST.LIST 8 DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

ORIGINAL PAGE IS
OF POOR QUALITY
C INTEGER INTTEMP 0 TEMPORARY
INTEGER KNTEMP 0 TEMPORARY

C PROCEDURE
C
C CALL TRACE
C
C ANY SPECIFICATIONS FROM USER?
C
KNTEMP='HUL'
CALL GETSK(HTEMP,'31', NULCST)
1F(KTYPE.EQ.'HUL') GO TO 800  & NO SPECS. SO CONFIRM CURRENT ONES
C
C CHECK GEOMETRY
C
1F(KTEMP.NE.'ERT').AND.
  & (KTEMP.NE.'HOM').AND.
  & (KTEMP.NE.'LCC').AND.
  & (KTEMP.NE.'PS').AND.
  & (KTEMP.NE.'SOM').AND.
  & (KTEMP.NE.'UTM').& HTEMP='BAD'
  & (KTEMP.NE.'BAD') CALL WARNS('BAD GEOMETRY ---')
  & (KTEMP.NE.'BAD') MERGED=KTEMP
  CALL GETSN(HTEMP, *1,-1,'EXTRA GEOMETRY SPECIFICATION ---')
C
C CONFIRM GEOMETRY
C
800 IF(NCFCHM.NE.0) WRITE(8,659) MEROED
   659 FORMAT(' GEOMETRY, ',A3)
C
KOND=''
RETURN
END
SUBROUTINE KMOHEA  
GET/CHECK PAGE HEADINGS
U KOMO1  & I: FIRST 3 CHAR'S OF COMMAND  O: SPACES

HISTORY
-------
E H SCHLOSSER  LEC  11/24/73  NUMERIC OPTION
E H SCHLOSSER  LEC  10/13/75  ALPHANUMERIC COMMAND
E H SCHLOSSER  LEC  07/17/78  DELETE RETURN K
E H SCHLOSSER  LEC  02/20/79  REVISE GETS.. CALLS
E H SCHLOSSER  LEC  11/14/79  UPGRADE DOCUMENTATION

METHOD
------
UPDATE HEADING LINES 1 & 2 FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE
----------------------
NONE.

EXTERNAL REFERENCES
-------------------
GETSKH   & GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN   & GET INTEGER DATA FIELD FROM UNIT 5

EXCEPTIONS
----------
1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE 'WARNING' DIAGNOSTICS:
   1 <= LINE NUMBER <= 2

2. ANY COMMA EMBEDDED IN THE HEADING SPECIFICATION TERMINATES THE HEADING -- THE REMAINDER IS IGNORED.

GLOBAL DECLARATIONS
--------------------
INCLUDE KOMXO1.LIST   & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST   & DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS
--------------------
INTEGER NH01,NH02   & LINE NUMBER OF 1ST, 2ND HEADING LINE TO CONFIRM
PROCEDURE
---------

CALL TRACE

GET/CHECK LINE NUMBER & HEADING TEXT

NHD1=0
CALL GETSIN(NHD1). 1.2, 'BAD HEADING LINE NUMBER (NOT 1 OR 2) --'
IF(NHD1.NE.0) GO TO 200
   NHD1=1
   NHD2=2
GO TO 300
200 NHD2=NHD1
CALL GETSKH(JHOG(JHOG(NH01).),72), NULCST)

CONFIRM HEADING LINE(S)

300 IF(MCFR.M.EQ.0) GO TO 900
   400 NHD=NHD1.NHD2
       WRITE(6,355) NHD.(JHOG(NHD,NHD).NHD=1.12)
355 FORMAT(' HEADING. ',11'. ',12AS)
   400 CONTINUE

COMMON RETURN

900 KOMO=''
   RETURN
END
SUBROUTINE KMDIF: CONDITIONALLY PERFORM NEXT BLOCK OF COMMANDS
U KOMDI: FIRST 3 CHAR OF COMMAND O: SPACES

HISTORY

E H SCHLOSSER LEC 09/20/74 ORIGINAL CODE IN KMONEX
E H SCHLOSSER LEC 07/24/78 DELETE RETN K. ADD LEGEND
E H SCHLOSSER LEC 12/05/78 CHANGE TO KMDIF
E H SCHLOSSER LEMS CO 12/05/78 CHG MAX #CMDS IN BLOCK FROM 12 TO 20

METHOD

COMPARE SETTING OF CONDITION SWITCH WITH SETTING OF SPECIFIED MODE.
IF BOTH SETTINGS ARE NOT THE SAME, SKIP ALL COMMANDS UNTIL THE
NEXT F1 COMMAND.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSKH 0 GET SPECIFICATION FROM UNIT 5
WARNS 0 FLAG LAST SPECIFICATION FROM UNIT 5 AS INVALID
GETSIN 0 GET INTEGER DATA FIELD FROM UNIT 5
READS 0 READ NEXT COMMAND INTO UNIT 5 BUFFER

EXCEPTIONS

1. INVALID CONDITION & MODE SPECIFICATIONS ARE DETECTED & FLAGGED.
2. A NESTED 'IF' COMMAND GENERATES A 'WARNING' DIAGNOSTIC.
3. IF MORE THAN 20 COMMANDS INTERVENE BETWEEN THE CURRENT 'IF' COMMAND
AND THE NEXT 'FI' COMMAND, THEN THE BLOCK IS TERMINATED WITH A
'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KMONEX.LIST 0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST 0 DEFINE NULL CHARACTER STRING
INCLUDE MAXINT.LIST 0 DEFINE MAXIMUM INTEGER

M-45
C LOCAL DECLARATIONS
C

INTEGER KOND = CONDITION SPECIFIED BY USER ('OFF' OR 'ON')
INTEGER KONOSW = CONDITION SWITCH SETTING (0 OR 1 IF VALID)
INTEGER MODE = MODE SPECIFIED BY USER
INTEGER MODESW = MODE SWITCH SETTING (0 OR 1 IF VALID)
INTEGER INTEMP = TEMPORARY
INTEGER NSKIP = NUMBER OF COMMANDS SKIPPED
INTEGER KNSKP = NAME OF COMMAND BEING SKIPPED

C PROCEDURE
C

CALL TRACE

C GET/CHECK CONDITION 'OFF' OR 'ON' & CONVERT TO CONDITION SWITCH

KOND=' NUL'
CALL GETSKM(KOND.(3), NULCST)
KONOSW=MAXINT
IF(KOND.EQ.'OFF') KONOSW=0
IF(KOND.EQ.'ON ' KONOSW=1
IF(KONOSW.EQ.MAXINT) CALL WarNS( 'BAD CONDITION (NOT ON OR OFF) --')

C GET/CHECK MODE & FIND CURRENT VALUE OF ITS MODE SWITCH

MODE=' NUL'
CALL GETSKM(MODE.(3), NULCST)
MODESW=MAXINT
IF(MODE.EQ.'BAT') MODESW=MINO(MBATCH.1) BATCH
IF(MODE.EQ.'CHE') MODESW=MINO(MCHECK.1) CHECKOUT
IF(MODE.EQ.'COL') MODESW=MINO(MCOLOR.1) COLOR
IF(MODE.EQ.'CON') MODESW=MINO(MCONFIRM.1) CONFIRM
IF(MODE.EQ.'DUM') MODESW=MINO(MDUMP.1) DUMP
IF(MODE.EQ.'ECM') MODESW=MINO(MECO.1) ECHO
IF(MODE.EQ.'LEO') MODESW=MINO(MLEGEND.1) LEGEND
IF(MODE.EQ.'PRD') MODESW=MINO(MPROMPT.1) PROMPT
IF(MODE.EQ.'TRA') MODESW=MINO(MTRACE.1) TRACE
IF(MODESW.EQ.MAXINT) CALL WarNS( 'BAD MODE --')
CALL GETSKM(INTEMP. *1,-1,'EXTRA IF SPECIFICATION --')

C IF SPECIFICATION VALID & KONOSW <> MODESW, THEN SKIP COMMANDS INSIDE BLOCK

IF((KONOSW.EQ.MAXINT).OR. KCOND INVALID)
6 (MODESW.EQ.MAXINT).OR. MODE INVALID
6 (KONOSW.EQ.MODESW) MODE SWITCH IN SPECIFIED CONDITION
6 GO TO 900 THEN DON'T SKIP COMMANDS IN BLOCK:!!
DO 800 NSKIP=1.21

M-48
CALL READS(LSSTAT, NULCST)
CALL GETS(KHDSKP,(3), 'NULCST')
IF(KHDSKP.EQ.' IF ') CALL MDWARNI('NESTED IF NOT ALLOWED')
   IF(KHDSKP.EQ.'FI ') GO TO 900
800 CONTINUE
   CALL MDWARNI('BAD IF...FI BLOCK -- MORE THAN 20 COMMANDS OR MISSING FI')
C
C NORMAL RETURN
C
900 KOMDS =
   RETURN
   END
SUBROUTINE KNDINT  & GET/CHECK INTENSITY
U KOND)  & I: FIRST 3 CHAR OF COMMAND O: SPACES

HISTORY

E H SCHLOSSER LEC 05/28/79 MODIFY FROM KMDSYM
E H SCHLOSSER LENSCO 05/30/80 SUPPORT ENCODED PERCENTS

METHOD

GET, CHECK, INTERPOLATE IF NECESSARY, AND STORE THE
INTEGER-INTENSITY-EQUIVALENTS (I-I-E) FOR CRT PICTURES IN
CHARACTER S OF THE KSYM ARRAY ELEMENTS.

MACHINE-DEPENDENT CODE

ASSUMES AT LEAST 5 CHARACTERS PER KSYM ELEMENT.

EXTERNAL REFERENCES

GETSIN  & GET INTEGER DATA FIELD FROM UNIT S
OCODE  & DECODE NUMERIC CHARACTER STRING
WARNS  & GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT S
UNGETS  & BACK UP ONE DATA FIELD ON UNIT S
MDWARN  & PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
PUTICE  & PUT INTEGER-CHARACTER-EQUIVALENT INTO CHARACTER STRING
OETICE  & OET INTEGER-CHARACTER-EQUIVALENT FROM CHARACTER STRING

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   COLOR MODE SWITCH (MCOLOR IN KMXQT) NOT ON

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   .05 < INTENSITY < 1.00
   0 < NUMBER < ISYMNI (SEE KOMSYM-PROC)

3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KMXQT.LIST  & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMSYN.LIST  # COMMON SYMBOL TABLE
INCLUDE NULCST.LIST  # DEFINE NULL CHARACTER STRING
INCLUDE MAXINT.LIST  # DEFINE MAXIMUM INTEGER

LOCAL DECLARATIONS

NOTE: THE 'MINIMUM' INTENSITY, I-1-E, & NUMBER DECLARED BELOW PERTAIN TO THE FIRST INTENSITY & NUMBER SPECIFIED BY THE USER. THE 'MAXIMUM' INTENSITY, I-1-E, & NUMBER DECLARED BELOW PERTAIN TO THE SECOND INTENSITY & NUMBER SPECIFIED BY THE USER. THE ACTUAL VALUES OF THE 'MINIMUM' I-1-E & NUMBER ARE NOT NECESSARILY LESS THAN THE VALUES OF THE 'MAXIMUM' I-1-E & NUMBER.

INTEGER NOSAVE  # TEMPORARY SAVE FOR CONTENTS OF NOTOTL ON ENTRY
INTEGER INTTEMP  # TEMPORARY
INTEGER KHTEMP  # TEMPORARY
INTEGER MINPCT, MAXPCT  # MINIMUM, MAXIMUM INTENSITY PERCENT
INTEGER MINIE, MAXIE  # MINIMUM, MAXIMUM INTENSITY-EQUIVALENT
REAL AMINII  # 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)

PROCEDURE

CALL TRACE

INITIALIZE

IF(MCOLOR.NE.0) GO TO 120
CALL MDWARNI
"'INTENSITY COMMAND NOT ALLOWED (COLOR MODE NOT ON)'")
GO TO 900

120 NOSAVE=NOTOTL

GET OPTIONAL 'MINIMUM' INTENSITY (I) FROM 1ST SPEC FIELD

WHILE *' NUL'
CALL GETSKH(KHTEMP,4), NULCST1
IF(KHTEMP.EQ.4' NUL') GO TO 850  # NO SPECS, SO CONFIR EVERYTHING
LOCPTC=CHREC(KHTEMP,1,4,9)
KODTYP='ERR'
IF(LOCPTC.0T.1) CALL DCODE(INTEMP,RTTEMP,KODTYP.
"KHTEMP,(1),(LOCPTC-11)
IF(INTEMP.LT.1) KODTYP='ERR'
IF(INTEMP.0T.100) KODTYP='ERR'
IF(KODTYP.EQ.4'IN') MINIIE=(INTEMP-1)/10
IF(KODTYP.NE.4'IN') CALL WARNI "BAD FIRST INTENSITY --'

M-49
DAM PACKAGE APPENDIX M
COMMAND ROUTINES

GET REQUIRED 'MINIMUM' NUMBER (INTEGER) FROM 2ND SPEC FIELD

MINNUM=MAXINT
CALL GETSIN(MINNUM, 0,SYMHI, 'BAD FIRST NUMBER --')

INITIALIZE IMPLICIT 'MAXIMUM' INTENSITY AND NUMBER

MAXIIE=MINIIE
MAXNUM=MINNUM

IS 3RD SPEC FIELD INTEGER?

INTEP=MAXINT
CALL GETSIN(INTEP, -9999.9999, NULCST) & STILL MAXINT IF NOT INTEGER
CALL UNGETS

IF 3RD SPEC FIELD NOT INTEGER, GET OPTIONAL 'MAX' INTENSITY (I) FROM IT

KHTEMP= 'NUL'
IF(INTEP.EQ.MAXINT) CALL GETSKH(KHTEMP,(4), NULCST)
IF(KHTEMP.EQ. 'NUL') GO TO 220
LOCPC=LOCPCX(KHTEMP,(4), (4), 'I')
KOTYP='ERR'
IF(LOCPC=LOCPC-1) CALL DCODE(INTEP, LTEMP, KOTYP, KHTEMP,(4), (4), (4), 'I')
KHTEMP=LOCPCX(KHTEMP,(4), (4), (4), 'I')
IF(INTEP.LT.1) KOTYP='ERR'
IF(INTEP>O.100) KOTYP='ERR'
IF(KOTYP.EQ. 'IN') MAXIIE=(INTEP-1)/10
IF(KOTYP.NE. 'IN') CALL WARNS('BAD SECOND INTENSITY --')
220 CONTINUE

GET OPTIONAL 'MAXIMUM' NUMBER (INTEGER) FROM NEXT SPEC FIELD

CALL GETSIN(MAXNUM, 0,SYMHI, 'BAD SECOND NUMBER --')

COMPUTE INTENSITIES PER NUMBER

INC1IE=1
IF(MINIIIE.OE.MAXIIE) INC1IE=1
MAXNUM=M110(MAXNUM, SYMHI)
INCNUM=1
IF(MINNUM.OE.MAXNUM) INCNUM=1

TNUM=FLOAT(MAXIIE-MINIIIE+INC1IE)/ INCNUM  & PREPARE TO INTERPOLATE ...
& FLOAT(MAXNUM-MINNUM+INCNUM) & ... OVER INCLUSIVE RANGE
IF(AABS(TNUM).OE.1.) CALL MDWARN('MORE THAN 1 INTENSITY:PER NUMBER')

CHECK FOR EXTRA SPEC FIELDS & FOR DIAGNOSTICS
C

CALL GETSIN(INTEMP, +1, -1, 'EXTRA INTENSITY SPECIFICATION --')
IF(NOSAVE.NE.NOTOTL) GO TO 900

C INTERPOLATE 1-1-E'S & LOAD INTO INTENSITY PART (CHAR 9) OF SYMBOL TABLE
C
AMINIE=FLOAT(MINIE)+0.0001  & INTERPOLATE UP FROM LOW SIDE OF MINIE
IF(MINIE.GT.MAXIE) DO 300 NUM=MINNUM.MAXNUM.1NCNUM
      CALL PUTICE(KSYM(NUM+1),(S). & INTERPOLATE 1-1-E
      IFIX(AMINIE+FLOAT(NUM-MINNUM-TNINUM))
 300 CONTINUE

C CONFIRM INTENSITIES
C
MINPCT=(MINIE+1)*10
MAXPCT=(MAXIE+1)*10
IF(MCFIRM.NE.0) WRITE(S,329) MINPCT.MINNUM.MAXPCT.MAXNUM
329 FORMAT(' INTENSITY. '.14.'i'.'.14.'i'.'.J3.')
GO to 900

C NO INTENSITIES SPECIFIED -- CONFIRM ALL INTENSITIES
C
950 IF(MCFIRM.NE.0) CALL ALLINT
C
NORMAL RETURN
C
900 KMOD= ' RETURN
C
C INTERNAL SUBROUTINE ALLINT & CONFIRM ALL CURRENT INTENSITIES
C
INTEGER NXT11E & NEXT 1-1-E
C
MINNUM=0
CALL GETICE(MINIE,KSYM(1),(S))
DO 200 MAXNUM=0,1SYM
      CALL GETICE(NXT11E,KSYM(MAXNUM+2),(S))
      IF(NXT11E.EQ.MINIE) GO to 200 & NEXT 1-1-E SAME AS CURRENT
      MINPCT=(MINIE+1)*10
      IF(MINNUM.EQ.MAXNUM)
      WRITE(S,145) MINPCT.MINNUM
145 FORMAT(' INTENSITY. '.14.'i'.'.J3')
      IF(MINNUM.NE.MAXNUM)

M-51
DAM PACKAGE APPENDIX M
COMMAND Routines

KNDINT 000

WRITE(*,165) MINPCT,MINNUM,MAXNUM
165 FORMAT(' INTENSITY,'.14,'E',"..",J3,'..',J3)
MINNUM=MAXNUM+1
MINTIE=NXTIE
200 CONTINUE
RETURN
C
C
END
SUBROUTINE KMONLY 8 GET/CHECK LINEAR HEIGHTS/GAIN/BIAS
U KOND1 8 1: FIRST 3 CHAR OF COMMAND 0: SPACES

MACHINE-DEPENDENT CODE

SCALE INTEGER BIASES & WEIGHTED GAINS ASSUME 36-BIT INTEGERS.
DIMENSION AND FORMAT SPECIFICATIONS ASSUME 6 CHARACTERS PER INTEGER.

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   COEFFICIENT TYPE NOT WEIGHTS) OR GAINS) OR BIAS(S)

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   1 <= CHANNEL <= 2
   -10 <= WEIGHT <= +10
   -10 <= GAIN <= +10
   -100 <= BIAS <= +100

3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.
GLOBAL DECLARATIONS

INCLUDE KOMKOT.LIST
INCLUDE KOMNER.LIST
INCLUDE KOMKRT.LIST
INCLUDE NULCST.LIST

LOCAL DECLARATIONS

REAL TBIAS(2) 8 TEMP STORAGE FOR BIASES AS REAL NUMBERS
INTEGER JFMT(8) 8 DYNAMIC FORMAT -- JFMT(8) CHANGES

DATA JFMT/*(9H LINEAR, 11H WEIGHTS, 00000012H, .FB.41)**/
C GET/CHECK/CONFIRM GAIN
C 400 IF ITTEM.EQ."GAIN") GO TO 900
CALL GET5RL(RTLGAIN(NL)) 1,..10,..10.."BAD GAIN --"
IF(INCFIRN.NE.0) WRITE(8,425) NL,RTLGA1111NL)
425 FORMAT('LINEAR. ',11,.' GAIN. ',-93)GO TO 700
C C GET/CHECK/CONFIRM BIAS
C 500 IF ITTEM.EQ."BIA") GO TO 900
CALL GET5RL(TBIAS(NL)) 1,..100,..100.."BAD BIAS --"
LRTB1E(NL)=TBIAS(NL)*2.012
IF(INCFIRN.NE.0) WRITE(8,525) NL,TBIAS(NL)
525 FORMAT('LINEAR. ',11,'. BIAS. ',-93)GO TO 900
C C UPDATE WEIGHTED GAINS
C 700 DO 750 NCH=1.NERCHA
DO 730 NL=1.2
LRTWH(NCH,NL)=RTLWOT(NCH,NL)*RTLGA1111NL)*2.012
730 CONTINUE
750 CONTINUE
GO TO 100
C C FLAG BAD SPECIFICATION
C 800 IF ITTEM.EQ."NONE") GO TO 900
CALL WARN(S( 'BAD LINEAR SPECIFICATION --"
C C C CONFIRM WEIGHTS/GAINS/BIASES
C 900 IF(INCFIRN.EQ.0) GO TO 900
DO 800 NL=1.2
CALL CSTRN(JFMT(8),(11,16), NERCHA, (8))
WRITE(8,JFMT) NL,RTLWOT(NCH,NL), NCH=1, NERCHA
WRITE(8,425) NL,RTLGA1111NL:
WRITE(8,525) NL,TBIAS(NL)
800 CONTINUE
C C C NORMAL RETURN
C 900 KOND=' RETURN
RETURN
END
SUBROUTINE KNOHER 2 GET/CHECK TRANSVERSE MERCATOR CENTRAL MERIDIAN

U KOND' 6 1: FIRST 3 CHAR OF COMMAND 61 SPACES

METHOD

UPDATE CENTRAL MERIDIAN FROM UNIT 9. IF SPECIFIED, AND CONFIRM.

EXTERNAL REFERENCES

GETSSX 8 GET SEXAGESARY DATA FIELD FROM UNIT 9
GETSIN 8 GET INTEGER DATA FIELD FROM UNIT 9

EXCEPTIONS

1. SPECIFICATION OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   50. < MERIDIAN < 180.

2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KOMKOT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMFIT.LIST 8 COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMOMW.LIST 8 COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDEF.LIST 8 DEFINE STRUCTURE OF WINDOW PACKETS

LOCAL DECLARATIONS

REAL UTDCHD 8 VALUE OF UDCHD ON ENTRY (BEFORE POSSIBLE CHANGING)
INTEGER INTEMP 8 TEMPORARY
CALL TRACE

IF(UTMCMO.EQ.0.) UTMCMO=36000.  ! INVALID
OLDCMD=UTMCMO
CALL GETSSX(UTMCMO, 1..50..180.."BAD MERIDIAN --")
IF(MCFIRM.NE.0) WRITE(*,125) UTMCMO
125 FORMAT(" MERIDIAN, "F9.4," DEGREES CENTER OF TM PROJECTION")
CALL GETSIN(INTEMP, 1..1.."EXTRA MERIDIAN SPECIFICATION --")
IF(OLCCMO.NE.UTMCMO) KSTOWM(WORG)="NUL"  & IF CMD CHANGED MARK ORIG ABSENT
900 KOMD="'
RETURN
END
SUBROUTINE KMONAM: GET/CHECK NAME OF TRANSFORMATION OR SPECTRAL LIMITS
U KOMD) & 1: FIRST 3 CHARS OF COMMAND 0: SPACES
C
C HISTORY
C
C
C E H SCHLOSSER LEC 12/17/75 ORIGINAL CODE
C E H SCHLOSSER LEC 07/22/78 DELETE RETURN K
C E H SCHLOSSER LEC 02/20/79 REVISE GETS... CALLS
C E H SCHLOSSER LEC 11/29/79 UPGRADE DOCUMENTATION

C METHOD
C
C UPDATE NAME FROM UNIT S, IF SPECIFIED, AND CONFIRM.
C
C MACHINE-DEPENDENT CODE
C
C NONE.
C
C EXTERNAL REFERENCES
C
C GETSKH & GET CHARACTER STRING DATA FIELD FROM UNIT S
C GETSIN & GET INTEGER DATA FIELD FROM UNIT S
C
C EXCEPTIONS
C
C 1. IF LONGER THAN 24 CHARACTERS, THE USER-SPECIFIED NAME IS TRUNCATED
C TO 24 CHARACTERS.
C
C 2. IF THE USER-SPECIFIED NAME CONTAINS A COMMA, THAT PART FOLLOWING
C THE COMMA IS TREATED AS AN EXTRA SPECIFICATION AND GENERATES A
C 'WARNING' DIAGNOSTIC.
C
C LOCAL DECLARATIONS
C
C INTEGER INTEMP & TEMPORARY
M-58
C

C PROCEDURE

C ---------

C

CALL TRACE

C

CALL GETSKN(MTHERAL,24), HULCST)
CALL GETSIN(INTEMP, -1,-1,'EXTRA NAME SPECIFICATION --')
IF(MCFIRM.NE.0) WRITE(6,125) (MTHERAL(N),N=1,4)
125 FORMAT(' NAME, ',A8)
KOND=''
RETURN
END
SUBROUTINE KMONEW (prints news of recent program changes)
U (kond) A: first 3 chars of command, 0: spaces

HISTORY

E H Schlosser LEC 12/19/75 Original code
E H Schlosser LEC 07/17/78 delete return k
E H Schlosser LEC 02/20/79 revise gets.. calls
E H Schlosser LEC 11/15/79 upgrade documentation

METHOD

For each user specification:
GET SPECIFICATION & APPEND TO 'DAM.NEW-'
CALL RITELT TO WRITE DISK SYMBOLIC ELEMENT WITH ABOVE NAME.

MACHINE-DEPENDENT CODE

DIMENSION SPECIFICATIONS ASSUME 6 CHARACTERS PER WORD.
CONDITIONAL RETURNS TO STATEMENT LABELS

EXTERNAL REFERENCES

OETSKH i OETSKH CHARACTER STRING DATA FIELD FROM UNIT 5
PUTCHR i PUT CHARACTER INTO CHARACTER STRING
WARNS i GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
RITADD i WRITE OR $ADD DISK SYMBOLIC ELEMENT

EXCEPTIONS

1. If no news for the user specification is available, a 'warning'
diagnostic message is issued.

GLOBAL DECLARATIONS

INCLUDE NLCSLT.LIST i DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

INTEGER NAMELT(5) i NAME OF DISK SYMBOLIC ELEMENT CONTAINING NEWS
DATA (NAMELT(N),N=1,2) ' ' DAM.NEW-'

M-80
C

DAM PACKAGE APPENDIX M
COMMAND ROUTINES

C

C PROCEDURE
C -------
C
C CALL TRACE
C
C INITIALIZE 'DAM' SPEC BEFORE TRYING TO GET 1ST SPEC
C
NAMELT(3)='DAM'
C
C GET NEXT SPECIFICATION
C
100 CALL GETSKH(NAMELT(3),17,NULCST)
    CALL PUTCHR(NAMELT(3),18,'') & TERMINATE WITH BLANK
    IF(NAMELT(3).EQ.' NUL') GO TO 900 & NO MORE SPECS
C
C WRITE NEWS IF AVAILABLE
C
    CALL RITADD(B,NAMELT,0,$800)
C
C INITIALIZE NUL SPEC BEFORE TRYING TO GET NEXT SPEC
C
200 NAMELT(3)=' NUL'
    GO TO 100
C
C NEWS NOT AVAILABLE
C
800 CALL WARNS('BAD PROGRAM NAME --')
    GO TO 200
C
C COMMON RETURN
C
900 KOMD=' '
    RETURN
END

M-61
SUBROUTINE KMONEX: 8 CONDITIONALLY PERFORM NEXT COMMAND
U: KMONI: 8: FIRST 3 CHAR. OF COMMAND G: SPACES

---------------------------------------------------------------------

METHOD
-------

COMPARE SETTING OF CONDITION SWITCH WITH SETTING OF SPECIFIED MODE.
SKIP NEXT COMMAND UNLESS BOTH SETTINGS ARE THE SAME.

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

NONE.

EXTERNAL REFERENCES
-------------------

GETSKH  GET SPECIFICATION FROM UNIT 5
WARN   FLAG LAST SPECIFICATION FROM UNIT 5 AS INVALID
GETSIN  FLAG EXTRA SPECIFICATION

EXCEPTIONS
----------

1. INVALID CONDITION & MODE SPECIFICATIONS ARE DETECTED & FLAGGED.

GLOBAL DECLARATIONS
---------------------

INCLUDE KOMXOT.LIST
INCLUDE NULCST.LIST   DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS
-------------------

INTEGER KOND   8 CONDITION SPECIFIED BY USER ('OFF' OR 'ON')
INTEGER KONDSW 8 CONDITION SWITCH SETTING (0 OR 1 IF VALID)
INTEGER MODE   8 MODE SPECIFIED BY USER
INTEGER MODES   8 MODE SWITCH SETTING (0 OR 1 IF VALID)
PAM PACKAGE APPENDIX M
COMMAND ROUTINES

C PROCEDURE
C-----------------
C
C CALL TRACE
C
C GET/CHECK CONDITION 'OFF' OR 'ON' & CONVERT TO CONDITION SWITCH
C
KOND=' NUL'
CALL GETSKH(KOND,(3), NULCST)
KONOSW=-999
IF(KOND.EQ. 'OFF') KONOSW=0
IF(KOND.EQ. 'ON ') KONOSW=1
IF(KONOSW.EQ. -999) CALL WARN(S( 'BAD CONDITION (NOT ON OR OFF) --'
C
C GET/CHECK MODE & FIND CURRENT VALUE OF ITS MODE SWITCH
C
MODE=' NUL'
CALL GETSKH(MODE,(3), NULCST)
MODESW=-999
IF(MODE.EQ. 'BAT') MODESW=MINO(MBATCH,1) " BATCH
IF(MODE.EQ. 'CHE') MODESW=MINO(MCHECK,1) " CHECKOUT
IF(MODE.EQ. 'COL') MODESW=MINO(MCOLOR,1) " COLOR
IF(MODE.EQ. 'CON') MODESW=MINO(MCONFIRM,1) " CONFIRM
IF(MODE.EQ. 'DUM') MODESW=MINO(MDUMP,1) " DUMP
IF(MODE.EQ. 'ECH') MODESW=MINO(MECHO,1) " ECHO
IF(MODE.EQ. 'LEG') MODESW=MINO(MLEGEND,1) " LEGEND
IF(MODE.EQ. 'PRE') MODESW=MINO(MPROMPT,1) " PROMPT
IF(MODE.EQ. 'TRA') MODESW=MINO(MTRACE,1) " TRACE
IF(MODESW.EQ. -999) CALL WARN(S( 'BAD MODE --'
CALL GETSIN(ITEMP, +1,1, 'EXTRA SPECIFICATION --')
C
C SKIP NEXT COMMAND IF SPECIFICATION VALID & KONOSW NOT EQUAL MODESW
C
IF((KONOSW.0T. -999).AND. (KONOSW.EQ. MODESW)) CALL READ(S(STAT, NULCST)
C
C NORMAL RETURN
C
KOND=' 
RETURN
END
SUBROUTINE KNOFF  & TURN OFF MODE OPTIONS
U KOMD! & I: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY

E H SCHLOSSER  LEC  12/20/79  ORIGINAL CODE
E H SCHLOSSER  LEC  07/23/79  DELETE RETURN K
E H SCHLOSSER  LEC  02/13/79  REVISE GETS.. CALLS
E H SCHLOSSER  LEC  11/29/79  UPGRADE DOCUMENTATION

METHOD

FOR THE FIRST USER SPECIFICATION:
   IF 'DEFI UTI' TERMINATE PROCESSING OF DEFAULT COMMANDS.
   OTHERWISE CALL SETMOD TO SET & CONFIRM ALL SPECIFIED MODE
   SWITCHES OFF.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSKN  & GET CHARACTER STRING DATA FIELD FROM UNIT 5
UNGETS  & BACK UP ONE DATA FIELD ON UNIT 5
SETMOD  & SET & CONFIRM SPECIFIED MODE SWITCHES(S)
GETsin  & GET INTEGER DATA FIELD FROM UNIT 5

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   SPECIFICATION AFTER 'DEFI UTI'
   INVALID MODE SWITCHES(S) SPECIFIED (SEE SETMOD)

GLOBAL DECLARATIONS

INCLUDE KOMXGT.LIST  & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST  & DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

M-84

M-84
INTEGER KHTEMP  B TEMPORARY

PROCEDURE
-----

CALL TRACE

GET FIRST SPECIFICATION

KHTEMP=' NUL'
CALL GETSKM(KHTEMP,(3),-1,0)
IF(KHTEMP.EQ.'DEF') GO TO 400

SET SPECIFIED SWITCHES TO OFF (0)

120 IF(KHTEMP.NE.' NUL') CALL UNOETS
CALL SETM00(0)
GO TO 900

TERMINATE PROCESSING OF DEFAULT COMMANDS

400 IF(NMNOW.NE.0) GO TO 120
CALL MDFATL('INVALID DEFAULT COMMANDS')

WRITE(6,425)

425 FORMAT(4X,'# USER COMMANDS: ')"/

DONE

900 KMD="  RETURN
END
SUBROUTINE KNDON1 A TURN ON MODE OPTIONS
U KNDON1 A 1: FIRST 3 CHARS OF COMMAND O: SPACES

HISTORY

E H SCHLOSSER LEC 12/21/75 ORIGINAL CODE
E H SCHLOSSER LEC 07/23/78 DELETE RETURN K
E H SCHLOSSER LEC 02/13/79 REVISE GETS.. CALLS
E H SCHLOSSER LEC 11/29/79 UPGRADE DOCUMENTATION

METHOD

FOR THE FIRST USER SPECIFICATION:
IF 'DEFAULT' INITIATE PROCESSING OF DEFAULT COMMANDS.
OTHERWISE CALL SETMOD TO SET & CONFIRM ALL SPECIFIED MODE
SWITCHES ON.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSKH GET CHARACTER STRING DATA FIELD FROM UNIT 5
UNOGETS BACK UP ONE DATA FIELD ON UNIT 5
SETMOD SET & CONFIRM SPECIFIED MODE SWITCHES
GETSIN GET INTEGER DATA FIELD FROM UNIT 5

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   SPECIFICATION AFTER 'DEFAULT'
   INVALID MODE SWITCHES SPECIFIED (SEE SETMOD)

GLOBAL DECLARATIONS

INCLUDE KONKST.LIST A COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST A DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

M-66
DAN PACKAGE APPENDIX M
COMMAND ROUTINES

INTEGER KTEMP  8 TEMPORARY

PROCEDURE
----
CALL TRACE

GET FIRST SPECIFICATION

KTEMP=' NUL';
CALL GETSKN(KTEMP,11, NULCST)
IF(KTEMP.EQ. 'DEF') GO TO 400

SET SPECIFIED SWITCHES TO ON (1)

IF(KTEMP.NE. 'NUL') CALL UNGETS
CALL SETMOD(11) GO TO 900

INITIATE PROCESSING OF DEFAULT COMMANDS

400 IF(NWNDOW.NE.0) GO TO 120  8 DEFAULT CAN ONLY BE TURNED ON ONCE!!!
IF(NCARD.EQ.0) CALL MDFATL(
  'DEFAULT COMMANDS MAY NOT FOLLOW USER COMMANDS')
WRITE(6,425)
425 FORMAT(/'0 **CURRENT DEFAULT COMMANDS:'//)
CALL MDECLM1 NULCST)  8 CLEAR PREVIOUS WARNINGS
CALL GETSINS(KTEMP. *1.-1.'SPEC INVALID AFTER DEFAULT --')

DONE

900 KMOD= '
RETURN
END
SUBROUTINE KMDOORII & GET/CHECK ORIGIN OF VERTICES
U KOMDII & I: FIRST 3 CHAR OF COMMAND 0: SPACES

METHOD

GET USER-SPECIFIED COORDINATE SYSTEM & ORIGIN COORDINATES.
TRANSFORM INTO ORIGIN COORDINATES IN ALL OTHER COORDINATE SYSTEMS
EXCEPT PPD.

EXTERNAL REFERENCES

GETSKH 8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
WARNS 8 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
GETBIN 8 GET INTEGER DATA FIELD FROM UNIT 5
GETSIX 8 GET SEXAGEONARY DATA FIELD FROM UNIT 5
GETSRL 8 GET REAL DATA FIELD FROM UNIT 5
MDHARN 8 PRINT/COUNT/LOG "WARNING" DIAGNOSTIC MESSAGE
Q4A 8 GEOGRAPHIC FOR ADJUSTED MSS COORDINATES
U40 8 UTM FOR GEOGRAPHIC COORDINATES
A40 8 ADJUSTED MSS FOR GEOGRAPHIC COORDINATES
ALR41SX 8 REAL TO INTEGER SEXAGEONARY ARRAY
O4U 8 GEOGRAPHIC FOR UTM COORDINATES
A4P 8 ADJUSTED MSS FOR PRINT/PLT DEVICE COORDINATES

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   COORDINATE SYSTEM NOT SCAI10 OR DEGREES; OR KM OR METERS;
   COORDINATE SYSTEM KM OR METERS & UTM CENTRAL MERIDIAN
   (ZONE) NOT DEFINED.

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:

M-68
GLOBAL DECLARATIONS

INCLUDE KOMX0.LIST  COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST  COMMON ERTS SCENE PARAMETERS
INCLUDE KOMF0T.LIST  COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMIMW.LIST  COMMON INPUT WINDOW PACKETS
INCLUDE KOMOWW.LIST  COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDEF.LIST  DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE NULCST.LIST  DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

INTEGER INTTEMP(1), TEMPORARY
REAL ALTEMP(1), TEMPORARY
INTEGER KORDSY, COORDINATE SYSTEM: 'SCA'/'DEG'/'KM'/'MET'/'PRI'

PROCEDURE

CALL TRACE

GET/CHECK COORDINATE SYSTEM

KORDSY=KEYOWNW(MORIG), USE OLD COORDINATE SYS IF NONE SPECIFIED
CALL GETSKM(KORDSY,3), NULLCST
IF(KORDSY.EQ.'SCA') GO TO 300
IF(KORDSY.EQ.'DEG') GO TO 400
IF(KORDSY.EQ.'KM') GO TO 500
IF(KORDSY.EQ.'MET') GO TO 600
CALL WARNS('BAD COORDINATE SYSTEM --'), INVALID BUT TRANSFORM ANYWAY:"
GO TO 700

SCANNER (OSFC-ADJUSTED) COORDINATES

300 CALL OGETSIN(MSAOWW(MLIN,MORIG), -300.-3000.'BAD LINE --'),
CALL OGETSIN(MSAOWW(MSAM,MORIG), -200.-4000.'BAD SAMPLE --'),
320 CALL OGETSIN(MSAOWW(LAT,MORIG), OEDOWN(MLIN,MORIG),
FLOAT(MSAOWW(MLIN,MORIG)),
OEDOWN(MSAM,MORIG))
IF(NCFIRM.EQ.0) GO TO 800
WRITE(8,395) RSDOWNWLIN, HORIZ01, RSDOWNWSH, HORIZ01;
395 FORMAT(' ORIGIN. SCAN. '..'LINE. '..'SAMPLE')
WRITE(8,495) RSDOWNWLAT, HORIZ01, RSDOWNWLON, HORIZ01,
IF(UTCHD.EQ.0.0).1), RT(ABS(UTCHD),0.1).1) GO TO 800
CALL UTVRT(UVTOWNWEA, HORIZ01, UTHOWNWNO, HORIZ01),
* OEDOMNWLAT, HORIZ01, OEDOMNWLON, HORIZ01, UTHCHM;
WRITE(8,895) UTHOWNWEA, HORIZ01, UTHOWNWNO, HORIZ01
GO TO 800

C C GEOGRAPHIC COORDINATES (DEGREES)
C
400 CALL GETSSX(OEDOMNWLAT, HORIZ01),
* 1_..CTRLAT-S, CTRLAT+S, 'BAD LATITUDE ----'
CALL GETSSX(OEDOMNWLON, HORIZ01),
* 1_..CTRLON-S, CTRLON+S, 'BAD LONGITUDE ----'
CALL ANGRL(TENTP(1), RLTENTP(1), 1)
* OEDOMNWLAT, HORIZ01, OEDOMNWLON, HORIZ01,
MSAOWWLIN, HORIZ01, RLTENTP(1) = $ 8 ROUND IT!
MSAOWWSH, HORIZ01, RLTENTP(2) = $ 8 ROUND IT!
WRITE(8,895) RLTENTP(1), RLTENTP(2),
* OEDOMNWLAT, HORIZ01, OEDOMNWLON, HORIZ01,
CIF(NCFIRM.EQ.0.1) GO TO 800
WRITE(8,495) GEDOWNWLAT, HORIZ01, GEDOWNWLON, HORIZ01,
449 FORMAT(' ORIGIN. DEGREE..', F10.5, 'LATITUDE..', F10.5, '.LONGITUDE..'),
CALL RELAXG(OEDOMNWLAT, HORIZ01) = 0.0001, RLTENTP(1),
CALL RELAXG(OEDOMNWLON, HORIZ01) = 0.0001, RLTENTP(4),
WRITE(8,495) RLTENTP(1), RLTENTP(4),
449 FORMAT(' ORIGIN. DEGREE..', 14, ':', 'JE,' 'LATITUDE..', 14, ':', 'JE,' 'LONGITUDE..'),
WRITE(8,495) GEDOWNWLAT, HORIZ01, GEDOWNWLON, HORIZ01,
470 WRITE(8,895) MSAOWWLIN, HORIZ01, MSAOWWSH, HORIZ01
GO TO 800

C C UTM COORDINATES
C
600 CF=1.E-3 6 CONVERT FROM KILOMETRES TO METRES
GO TO 820
810 CF=1. 8 METRES -- NO CONVERSION NEEDED
820 IF(UTCHD.NE.0.1).AND.(ABS(UTCHD).LT.180.1) GO TO 830
CALL MWARN('NO UTM ZONE DEFINED')
GO TO 800
830 CALL GETSRL(UTHOWNWEA, HORIZ01), CF=0.1.E-6, 'BAD EASTING ----'
CALL GETSRL(UTHOWNWNO, HORIZ01), CF=0.1.E-6, 'BAD NORTHING ----'
CALL OEDOMNWLAT, HORIZ01, OEDOMNWLON, HORIZ01,
* UTHOWNWEA, HORIZ01, UTHOWNWNO, HORIZ01,
CALL ANGRL(TENTP(1), RLTENTP(1), 1)
* OEDOMNWLAT, HORIZ01, OEDOMNWLON, HORIZ01,
MSAOWWLIN, HORIZ01, RLTENTP(1) = $ 8 ROUND IT!
MSAOWWSH, HORIZ01, RLTENTP(2) = $ 8 ROUND IT!
IF(NCFIRM.EQ.0.0) GO TO 800
WRITE(8,895) UTHOWNWEA, HORIZ01, UTHOWNWNO, HORIZ01

R-70
645 FORMAT(' ORIGIN. KM. '-3P,F7.3.' EAST. '-F8.3.' NORTH')
WRITE(6,345) MSAOWW(WLIN,WORIO),MSAOWW(WSAM,WORIO)
GO TO 800

C PRINT/PLT DEVICE COORDINATES (FOR DEBUGGING ONLY!!!)
C
700 CALL OETSRL(PPDOWW(WLIN,WORIO), 1..1..6000.. 'BAD LINE --')
CALL OETSRL(PPDOWW(WCOL,WORIO), 1..1..6000.. 'BAD COLUMN --')
IF(IRFD.LE.0) GO TO 900  & SCALE NOT CALIBRATED -- CAN'T TRANSFORM PPD
CALL A4P(ADJLIN,ADJSAM,
    PPDOWW(WLIN,WORIO),+5,PPDOWW(WCOL,WORIO)+.5)
MSAOWW(WLIN,WORIO)=ADJLIN
MSAOWW(WSAM,WORIO)=ADJSAM
KOROSY='SCA' & USE SCANNER COORDINATES
GO TO 320

C STORE COORDINATE SYSTEM
C
800 KSYOWW(WORIO)=KOROSY
CALL OETSIN(INTEMP(1), +1, 'EXTRA ORIGIN SPECIFICATION -- ')

C
C NORMAL RETURN
C
900 KOMD=' '
RETURN
END
DAN PACKAGE APPENDIX M
COMMAND ROUTINES

SUBROUTINE KMOPAG1 & SKIP TO TOP OF NEXT PAGE AND WRITE MESSAGE
U KOMD} & I: FIRST 3 CHARS OF COMMAND O: SPACES

-----------------------------
C
C HISTORY
C-------
C
C	 E H SCHLOSSER	 LEC	 12/01/75	 ORIGINAL CODE
C	 E H SCHLOSSER	 LEC	 07/14/79	 DELETE RETURN K
C	 E H SCHLOSSER	 LEC	 02/20/79	 REVISe GETS.. CALLS
C	 E H SCHLOSSER	 LEC	 11/30/79	 UPGRADE DOCUMENTATION
C

C METHOD
C------
C
C	 GET SPECIFICATION. PAGE EJECT. AND PRINT SPECIFICATION.
C
C MACHINE-DEPENDENT CODE
C-------------------------
C
C DIMENSION & FORMAT SPECIFICATIONS ASSUME 6 CHARACTERS PER WORD.
C
C EXTERNAL REFERENCES
C---------------------
C
C	 GET5KM	 3 GET CHARACTER STRING DATA FIELD FROM UNIT 5
C	 GET5IN	 3 GET INTEGER DATA FIELD FROM UNIT 5
C	 HONUNIT	 3 PRINT PAGE HEADING
C
C EXCEPTIONS
C----------
C
C	 1. IF NO SPECIFICATION IS GIVEN. THEN NO ACTION IS TAKEN.
C	 2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC MESSAGE.
C
C GLOBAL DECLARATIONS
C---------------------
C
C	 INCLUDE NULCST.LIST	 3 DEFINE NUL CHARACTER STRING
C
C LOCAL DECLARATIONS
C---------------------
C
C	 INTEGER KHTEMP(6) 3 CHARACTER STRING TO WRITE ON TOP OF PAGE
C
C PROCEDURE
C---------
CALL TRAC
CALL NOUNIT(0,8)
KTEMP(1) = 'NUL'
CALL GET5KN(KTEMP,(48), NULCST)
IF(KTEMP(1).NE. 'NUL') WRITE(8,145) KTEMP
145 FORMAT(8X,8A8/
CALL GETSIN(KTEMP, -1,-1,'EXTRA PAGE SPECIFICATION --')
KMD= ' ' RETURN
END
SUBROUTINE KMOPEE: 'PEEK' AT VARIABLES IN LABELLED COMMON
U KOMD: 8 I: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY
-------
E M SCHLOSSER LEC 10/24/79 REQUIREMENTS
E M SCHLOSSER LEC 10/25/79 DESIGN/CODE/TEST

METHOD
-------
CHECK NAME OF COMMON, MINIMUM LOCATION, MAXIMUM LOCATION, INCREMENT, FOR
VALIDITY.
IF INVALID, ISSUE DIAGNOSTIC(S). ELSE PRINT PRINT CONTENTS OF REQUESTED
COMMON LOCATIONS IN ALL OF THE FOLLOWING FORMATS:
INTEGER, REAL, CHARACTER STRING

MACHINE-DEPENDENT CODE
----------------------
DIMENSION & FORMAT SPECIFICATIONS ASSUME 8 CHARACTERS PER WORD.

EXTERNAL REFERENCES
-------------------

EXCEPTIONS
-----------
1. THE LAST 2 SPECIFICATIONS ARE OPTIONAL. ALL SPECIFICATIONS PRESENT ARE
CHECKED FOR VALIDITY AND RESULT IN WARNING DIAGNOSTICS IF BAD.
2. KOMD IS NOT CHECKED FOR VALIDITY.

GLOBAL DECLARATIONS
---------------------
COMMON/KOMALT/KOMALT(1) & COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS
COMMON/KOMDET/KOMDET(1) & COMMON DETECTION FILE WINDOW PKTS & DATES
COMMON/KOMFIT/KOMFIT(1) & COMMON ADJUSTMENT/REGISTRATION PARAMETERS
COMMON/KOMIR/KOMIR(1) & COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
COMMON/KOMLW/KOMLW(1) & COMMON INPUT WINDOW PACKETS
COMMON/KOMKLS/KOMKLS(1) & COMMON CLASSIFICATION INFO
INCLUDE KOMLOG.LIST  COMMON LOG FILE BUFFER, I/O PKT. POINTERS
COMMON/KOMLUE/KOMLUE(1)  COMMON POINTERS/FLAGS FOR UNIT 2
COMMON/KOMLUE/KOMLUE(1)  COMMON POINTERS/FLAGS FOR UNIT 3
COMMON/KOMLUE/KOMLUE(1)  COMMON POINTERS/FLAGS FOR UNIT 5
COMMON/KOMLUE/KOMLUE(1)  COMMON I/O PACKETS FOR DETECTION FILES (21-24)
COMMON/KOMNER/KOMNER(1)  COMMON ENTS SCENE PARAMETERS
COMMON/KOMNET/KOMNET(1)  COMMON CONTROL NETWORK COORDINATES
COMMON/KOMNOW/KOMNOW(1)  COMMON OUTPUT WINDOW PACKETS
COMMON/KOMSLM/KOMSLM(1)  COMMON SPECTRAL LIMITS
COMMON/KOMSYM/KOMSYM(1)  COMMON SYMBOL TABLE
COMMON/KONTBL/KONTBL(1)  COMMON MULTI-PURPOSE TABLE
INCLUDE KONXOT.LIST  COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST  DEFINE NULL CHARACTER STRING

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 PROCEDURE

CALL TRACE

C GET NAME OF COMMON

CALL MDCLRH(NULCST)
NAMCOM=' ???'
CALL GET5KH(NAMCOM(61), NULCST)

C GET/CHECK LOCATION OF FIRST VARIABLE TO 'PEEK' AT

LOCBEG=0
CALL GET5IN(LOCBEG, 1.2500.'**BAD PEEK LOCATION --')

C GET/CHECK LOCATION OF LAST VARIABLE TO 'PEEK' AT

LOCEND=LOCBEG  A DEFAULT IF NOT SPECIFIED
CALL GET5IN(LOCEND, LOCBEG,LOCBEG+250.'**BAD PEEK END --')

C
C GET/CHECK INCREMENT

LOCINC=1  A DEFAULT IF NOT SPECIFIED

M-75
CALL GETSIN(LOCINC, +1, 0, 'BAD PEEK INCREMENT -- ')  
CALL GETSIN(LOCINC, +1, -1, 'EXTRA PEEK SPECIFICATION --')

C GIVE UP IF ANY WARNINGS OCCURRED
IF (NDWARN.NE.0) GO TO 900

C FIND REQUESTED COMMON & PRINT
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. 'KOMIR') CALL PEEK(KOMIR, KOMIR)
IF (NAMCOM.EQ. 'KOMKLS') CALL PEEK(KOMKLS, KOMKLS)
IF (NAMCOM.EQ. 'KOMLOG') CALL PEEK(KOMLOG, KOMLOG)
IF (NAMCOM.EQ. 'KOMLU') CALL PEEK(KOMLU, KOMLU)
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. 'KOMLOG') CALL PEEK(KOMLOG, KOMLOG)
IF (NAMCOM.EQ. 'KOMNET') CALL PEEK(KOMNET, KOMNET)
IF (NAMCOM.EQ. 'KOMSLM') CALL PEEK(KOMSLM, KOMSLM)
IF (NAMCOM.EQ. 'KOMSYM') CALL PEEK(KOMSYM, KOMSYM)
IF (NAMCOM.EQ. 'KOMXGT') CALL PEEK(KOMXGT, KOMXGT)
IF (.NOT. PEEKED) CALL NDWARN;
* 'BAD PEEK COMMON NAME -- -- "-, ' CBS4CS(NAMCOM(1), (6))

C DCNE

900 KOMD="":
CALL MOCLAW( NULCST)
RETURN

C C C

C INTERNAL
SUBROUTINE PEEK; & PRINT VALUES OF COMMON LOCATIONS
& KOM. & LABELLED COMMON (INTEGER OR CHARACTER)
& COM & LABELLED COMMON (REAL)

C INTEGER KOM(1) & ARGUMENT
REAL COM(1) & ARGUMENT
REAL RTTEM & TEMPORARY
INTEGER NFMT & TYPE OF 'F' FORMAT TO PRINT RTTEM WITH

C PEEKED=.TRUE.
DO 300 LOCPRT=LOCBE0, LOCEND, LOCINC
RTTEM=COM(LOCPRT)+ZERO & FORCE NORMALIZATION IF NOT REAL

M-78
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,
   6(1X,99,COM,LOCPRT,KOM(LOCPRT),RLTEMP,KOM(LOCPRT))
   FORMAT(1X,A6,'(',J4,')',110,1X,14.2X,A6)
   IF(NFORMT.EQ.2) WRITE(6,
   6(1X,99,COM,LOCPRT,KOM(LOCPRT),RLTEMP,KOM(LOCPRT))
   FORMAT(1X,A6,'(',J4,')',110,1X,14.6X,A6)
   IF(NFORMT.EQ.3) WRITE(6,
   6(1X,99,COM,LOCPRT,KOM(LOCPRT),RLTEMP,KOM(LOCPRT))
   FORMAT(1X,A6,'(',J4,')',110,1X,14.12X,A6)
300 CONTINUE
RETURN
END
SUBROUTINE KHDPL0( & GET/CHECK PLOTTER SPECIFICATIONS
U KOND) & I: FIRST 3 CHAR'S OF COMMAND 0; SPACES

HISTORY
---
R E Narveson LEC 18/04/79 MODIFIED FROM KHDPI
E H Schlosser LEC 01/11/79 REWRITE & DOCUMENT

METHOD
------
?????

MACHINE-DEPENDENT CODE
------------------------
FORMAT STATEMENTS ASSUME 8 CHARACTERS PER WORD.

EXTERNAL REFERENCES
---------------------
GETSKH & GET/CHECK CHARACTER STRING FIELD
GETSR & GET/CHECK REAL FIELD
GETSIN & GET/CHECK INTEGER FIELD
WARN & OUTPUT WARNING DIAGNOSTIC FOR PREVIOUS FIELD

EXCEPTIONS
----------
1. INVALID FUNCTION NAMES ARE FLAGGED.
2. SPECIFICATIONS FOR ALL VALID FUNCTIONS EXCEPT 'DEVICE' ARE CHECKED.
3. NO OTHER CHECKS ARE MADE.

GLOBAL DECLARATIONS
----------------------
INCLUDE KOMXGT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMPLT.LIST & COMMON PLOT FILE/DEVICE CHARACTERISTICS
INCLUDE NULCST.LIST & DEFINE NULL CHARACTER STRING
DIMENSION KOMPLT(5) & ??????
5 WORD COMMON KOMPLT?
CONTENTS UNITS INCHES/LENGTH REAL??
INCHES/WIDTH   REAL

DEVICE   CHARACTER
'CAL'   CALCOMP
'GER'   GERBER

MODEL   INTEGER
NUMBER OF PENS   INTEGER

LOCAL DECLARATIONS

INTEGER NAMFUN   PLOTTER FUNCTION NAME (SHEET, DEVICE, MODEL, PENS)
PARAMETER MINPEN = 1   MINIMUM NO. OF PENS
PARAMETER MAXPEN = 3   MAXIMUM NO. OF PENS
PARAMETER XMINLG = 5.0   MINIMUM INCHES/LENGTH
PARAMETER XMAXLG = 99.0   MAXIMUM INCHES/LENGTH
PARAMETER XMINWH = 5.0   MINIMUM INCHES/WIDTH
PARAMETER XMAXWH = 99.0   MAXIMUM INCHES/WIDTH

PROCEDURE

CALL TRACE

CHECK IF COMMAND IS NOW LEGAL
IF(NWNDOW.LT.+1) GO TO 100
CALL MOWARN('PLOTTER COMMAND CANNOT BE USED AFTER FIRST WINDOW IS PROCESSED')
GO TO 900

GET/CHECK PLOTTER FUNCTION TO BE SPECIFIED
100 NAMFUN='NONE'
   CALL GETSK(NAMFUN,(3), NULCST)   & GET 1ST 3 CHARACTERS OF FUNCT NAME
   IF(NAMFUN.EQ.'SHE') GO TO 300   & SHEET
   IF(NAMFUN.EQ.'DEV') GO TO 400   & DEVICE
   IF(NAMFUN.EQ.'MOD') GO TO 500   & MODEL
   IF(NAMFUN.EQ.'PEN') GO TO 600   & PENS
   IF(NAMFUN.EQ.'NONE') GO TO 300

FLAG BAD FUNCTION & DRAIN SPECS
   CALL WARNS('BAD PLOTTER FUNCTION --')
150 NAMFUN='NONE'
   CALL GETSK(NAMFUN,(3), NULCST)
   IF(NAMFUN.NE.'NONE') GO TO 150

M-79
C GET/CHECK/CONFIRM SHEET UNITS/LENGTH/WIDTH

C

300 CALL GETSRL(KOMPLT(1).
  = 1.0.XMINLX,XMAXLX, 'BAD LENGTH --'
CALL GETSRL(KOMPLT(2).
  = 1.0.XMINWX,XMAXWX, '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 GET/CHECK/CONFIRM DEVICE TYPE MNEMONIC

C

400 CALL GETSKH(MNEMON,(8), NULCST)
IF(MNEMON.EQ. ' ') CALL WARNS(' 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 MODEL SPECIFICATION

C

500 WRITE(6,515)
515 FORMAT(5X,'MODEL MNEMONICS ARE NOT DEFINED.')
GO TO 900

C

C GET/CHECK/CONFIRM NUMBER OF PEN(S)

C

600 MALTH=MIND,MALTH,MOUTH)
INSURE MALTH = MOUTH
CALL GETSIN(MALTH, 0,MOUTH, '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 EXIT

C

900 KOMR = ' '
RETURN
END
SUBROUTINE KMDPO1I  & GET/ CHECK CONTROL/ CHECK POINT
U KMDPO 1: FIRST 1/2/3 CHAR OF COMMAND OR POINT 0: SPACES IF VALID

HISTORY

E H SCHLOSSER LEC 07/25/73 ORIGINAL CODE
E H SCHLOSSER LEC 12/20/75 ALPHANUMERIC COMMAND
E H SCHLOSSER LEC 07/17/78 DELETE RETURN K
E H SCHLOSSER LEC 01/27/79 MODIFY FOR MACRO COMMANDS
E H SCHLOSSER LEC 11/30/79 REVISE 04A/440/04U/U40 CALLS

METHOD

GET COORDINATE SYSTEMS AND COORDINATES & STORE IN LABELLED COMMON.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

MDFATL  & PRINT/COUNT/LOG 'FATAL' DIAGNOSTIC MESSAGE
MOWARN  & PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
UNGTS  & BACK UP I FIELD ON UNIT 5
GETSKH  & GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN  & GET INTEGER DATA FIELD FROM UNIT 5
GETSRL  & GET REAL DATA FIELD FROM UNIT 5
GETSSX  & GET SEXAGESINARY DATA FIELD FROM UNIT 5
WARNS  & GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
O4A  & GEOGRAPHIC COORDINATES FOR ADJUSTED MSS COORDINATES
A40  & ADJUSTED MSS COORDINATES FOR GEOGRAPHIC COORDINATES
O4V  & GEOGRAPHIC COORDINATES FOR UTM COORDINATES
U4V  & UTM COORDINATES FOR GEOGRAPHIC COORDINATES

EXCEPTIONS

1. ALL SPECIFICATIONS ARE CHECKED FOR ALLOWABLE VALUES.

GLOBAL DECLARATIONS

INCLUDE KOMXGT.LIST  & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMNER.LIST  & COMMON ERTS SCENE PARAMETERS
INCLUDE KOMFIT.LIST  & COMMON CONTROL ADJUSTMENT/REGISTRATION PARAMETER
INCLUDE KONNET.LIST  8 COMMON CONTROL NETWORK
INCLUDE WINDEF.LIST  8 DEFINE STRUCTURE OF WINDOW PACKETS
INCLUDE NULCST  8 DEFINE NULL CHARACTER STRING

C LOCAL DECLARATIONS

REAL RLTEMP(2)  8 TEMPORARY
INTEGER INTEMP  8 TEMPORARY
INTEGER INTTEMP  8 TEMPORARY
INTEGER JDESCR(9)  8 DESCRIPTION (9 CHARACTERS)
INTEGER NOSW  8 TYPE COORD PRESENT FOR QUERY PT: 'NUL'/'EARTH'/'SCAN'

C PROCEDURE

CALL TRACE

C CHECK IF EXPLICIT 'POINT' COMMAND (KOMO IS P01)

IF(KOMO.EQ.'P').OR.
  & (KOMO.EQ.'PO').OR.
  & (KOMO.EQ.'PO1')) GO TO 110

C CHECK IF IMPLICIT 'POINT' COMMAND (KOMO IS POINT NUMBER)

CALL UNOETS
INTEMP=0  8 NOT VALID POINT NUMBER
CALL GETSIN(INTEMP, -99999, -99999, NULCST)
IF(INTEMP.EQ.0) GO TO 999
CALL UNOETS

C INITIALIZE

110 IF(NETH1.LT.350) GO TO 120
  CALL MORDALT ('MORE THAN 350 POINTS IN NETWORK')
  GO TO 900
120 NETH1=NETH1 + 1  8 SET NODE POINTER TO NEXT AVAILABLE NODE
NETP=1  8 NOT A QUERY POINT
NOSAVE=NOSAVE + 1

C POINT NUMBER

NETP(NETH1)=999999
CALL GETSIN(NETP(NETH1), -99999, -99999, 'BAD POINT NUMBER --
IF(NETP(NETH1).EQ.0) CALL WARN1 ('BAD POINT NUMBER --
IF(NETP(NETH1).GE.-999) GO TO 300
IF(NWINDW.OT.01) GO TO 250
CALL MOWARN1 ('NETWORK NOT YET ADJUSTED')
GO TO 300

H-82
290 IF [INCTRL.LT.8] CALL HMWARN('FEWER THAN 8 CONTROL POINTS')
   IF [INCTRL.LT.40] CALL HMWARN('LESS THAN 40% COVERAGE')
   IF [RMSNET.GT.150] CALL HMWARN('AND ERROR > 150 METERS')

C

C SCANNER (GSFC-ADJUSTED) COORDINATES

C 300 KTEMP='ABSENT'
   CALL GETSBN(KTEMP,1), NULCST)
   IF (KTEMP.EQ. 'S') OR (KTEMP.EQ. 'E') GO TO 300  & 'SCAN'
   CALL UGETS & SO WE CAN GET FIELD AGAIN WITH A DIFFERENT FORMAT
   IF (NETPT(NETT).OE. -999) GO TO 350 & NOT A QUERY POINT
   NOST='EARTH' & EARTH COORDINATES GIVEN FOR QUERY POINT
   IF (KTEMP.GT.0) GO TO 340 & ALPHAN COORDINATE SYSTEM (LINE & SAMPLE OMITTED)
   NOST='SCAN' & SCANNER COORDINATES GIVEN FOR QUERY POINT

320 CALL GETSRL(ADJNET,MLIN.NETT1).
   = 1...1500...500...*BAD LINE --'1 & REALS OK
   CALL GETSRL(ADJNET,MSAM.NETT1).
   = 1...0...500...*BAD SAMPLE --'1 & REALS OK
   IF (NETPT(NETT).LT. -999).AND.(NOSAVE.NE.NOTOTL) GO TO 800

C

C CHECK EARTH COORDINATE SYSTEM

C 340 KORDSY='ABSENT'
   CALL GETSKN(KORDSY,1), NULCST)
   IF (KORDSY.EQ. 'DEG') GO TO 400
   IF (KORDSY.EQ. 'KM ') GO TO 800
   IF (KORDSY.EQ. 'M') GO TO 610
   CALL HMWARN('BAD EARTH COORDINATE SYSTEM --')
   GO TO 800

C

C GEOGRAPHIC COORDINATES (DEGREES)

C 400 IF (NOST.EQ. 'SCAN') GO TO 470 & QUERY POINT, SCANNER COORD GIVEN
   CALL GETSBN(GEODET,MLAT.NETT1), 1.20..60...*BAD LATITUDE --')
   CALL GETSBN(GEODET,MLON.NETT1), 1.60..180...*BAD LONGITUDE --')
   IF (NOSAVE.NE.NOTOTL) GO TO 800
   IF (NOST.EQ. 'EARTH') GO TO 480 & QUERY POINT, EARTH COORD GIVEN
   IF (MCFIRM.EQ. '0') GO TO 900

430 CALL GETDES
   WRITE(6,448)
   NETPT(NETT1),
   ADJNET(MLIN.NETT1),ADJNET(MSAM.NETT1),
   GEODET(MLAT.NETT1),GEODET(MLON.NETT1),JDESC

448 FORMAT:
   6', POINT...O9,
   ', SCAN...FB.2',',FB.2,
   ', DEG...F10.5',',F10.3,A3,B8)
   GO TO 900

470 CALL GNA(GEODET,MLAT.NETT1),GEODET(MLON.NETT1),
   ADJNET(MLIN.NETT1),ADJNET(MSAM.NETT1),
   GO TO 430

480 CALL ANG(ADJNET,MLIN.NETT1),ADJNET(MSAM.NETT1),
   GEODET(MLAT.NETT1),GEODET(MLON.NETT1)
GO TO 430

C
C UTM COORDINATES
C

800 CF=1E-3  8 CONVERT FROM KILOMETRES TO METRES
GO TO 820

810 CF=1.  8 METRES -- NO CONVERSION NEEDED
820 IF (UTMCND.EQ.0) CALL MDARNI ("NO UTM ZONE DEFINED")
    IF (NSW.EQ."SCAN") GO TO 870  8 QUERY POINT. SCANNER COORD GIVEN
    CALL GETSLR(ALTTEMP(1), CF, 0.1E+6, "BAD EASTING --")
    CALL GETSLR(ALTTEMP(2), CF, 9E+6, "BAD NORTHING --")
    IF (NSAVE.NE.NOTOL) GO TO 800
    CALL GND4GEDN(MLAT,NETHI).GDN(MLON,NETHI).
    + ALTEMP(1).ALTTEMP(2).UTMCND)
    IF (NSW.EQ."EARTH") GO TO 680  8 QUERY POINT. EARTH COORD GIVEN
    IF (NSW.EQ.0) GO TO 900

630 CALL GETDES
    WRITE (8,849)
    6 NETPT(NETHI).
    6 ADJNET(MLIN.NETHI).ADJNET(MLIN.NETHI).
    6 ALTTEMP(1).ALTTEMP(2).JDECR
    849 FORMAT
    & POINT. .IS.
    &. SCAN..FS.2. FS.2.
    &. KM..IP. FS.4. FS.4.
    GO TO 900

670 IF (NSAVE.NE.NOTOL) GO TO 800
    CALL GND4GEDN(MLAT,NETHI).GDN(MLON,NETHI).
    + ADJNET(MLIN.NETHI).ADJNET(MLIN.NETHI).
    CALL GND4GEDN(MLAT,NETHI).GDN(MLON,NETHI).
    + ALTTEMP(1).ALTTEMP(2).GDN(MLON,NETHI).GDN(MLON,NETHI).
    GO TO 630

880 CALL AN4(ADJNET(MLIN.NETHI).ADJNET(MLIN.NETHI).
    + GDN(MLON,NETHI).GDN(MLON,NETHI))
    GO TO 630

C C
C NOT A VALID CONTROL/CHECK POINT -- SET NODE POINTER BACK
C

900 NETHI=NETHI-1
GO TO 900

C C
C KEEP COUNT OF CONTROL POINTS
C

900 IF (NETPT(NETHI).LT.-999) GO TO 800  8 QUERY POINT
    IF (NETPT(NETHI).GT.0) NCTLP=NCTLP+1

C C
C NORMAL RETURN
C

990 KOMP-
999 RETURN
C
SUBROUTINE GETDES
  DO 150 NMD=1,9
    150 JOESCR(NMD)=''
    IF(MATCH.EQ.0) GO TO 900
    CALL GETSKN(JDESCR(1), (48), NULCST)
    IF(JDESCR(1).NE.'') JOESCR(1)=''
    CALL GETSN(INTEMP, +1.-1, 'EXTRA POINT SPEC --')
  900 RETURN

END
SUBROUTINE KNPOK  & 'POKE' AT (CHANGE) VARIABLE IN LABELLED COMMON
U KOMDI  & I: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY
-----
E H SCHLOSSER  LEC  10/24/79  REQUIREMENTS
E H SCHLOSSER  LEC  10/25/79  DESIGN/CODE/TEST

METHOD
------
CHECK NAME OF COMMON AND LOCATION FOR VALIDITY.
IF INVALID, ISSUE DIAGNOSTIC(S). ELSE CHANGE & PRINT CONTENTS
OF REQUESTED COMMON LOCATION IN ALL OF THE FOLLOWING FORMATS:
  INTEGER, REAL, CHARACTER STRING
A VARIABLE ENCLOSED IN QUOTES IS ALWAYS INTERPRETED AS
A CHARACTER STRING.

MACHINE-DEPENDENT CODE
----------------------
DIMENSION & FORMAT SPECIFICATIONS_ASSUME 6 CHARACTERS PER INTEGER.

EXTERNAL REFERENCES
-------------------
MDCLRW  & CLEAR 'WARNING' DIAGNOSTICS
GETSKH  & GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSN  & GET INTEGER DATA FIELD FROM UNIT 5
DCODE  & DECODE NUMERIC CHARACTER STRING
MDWARN  & PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
MOVCST  & MOVE CHARACTER STRING
       & INTEGER LCHREQ  & LOCATE CHARACTER EQUAL SEARCH CHARACTER
       & DOUBLE PRECISION C854CS  & VARIABLE-LENGTH CST FOR FIXED-LENGTH CST

EXCEPTIONS
----------
1. ALL SPECIFICATIONS ARE CHECKED FOR VALIDITY AND RESULT
   IN WARNING DIAGNOSTICS IF BAD.
2. KOMDI IS NOT CHECKED FOR VALIDITY.

GLOBAL DECLARATIONS
---------------------
COMMON/KOMALT/KOMALT(1)  & COMMON ALTERNATE PRINT FILE COUNTERS, POINTERS
COMMON/KOMDET/KOMDET(1)  & COMMON DETECTION FILE WINDOW PKTS & DATES
COMMON/KOMFIT/KOMFIT(11) COMMON ADJUSTMENT/REGISTRATION PARAMETERS
COMMON/KOMIRT/KOMIRT(11) COMMON IRRADIANCE TRANSFORMATION COEFFICIENTS
COMMON/KOMIWM/KOMIWM(11) COMMON INPUT WINDOW PACKETS
COMMON/KOMKLS/KOMKLS(11) COMMON CLASSIFICATION INFO
INCLUDE KOMLOG.LIST COMMON LOG FILE BUFFER, I/O PKT. POINTERS
COMMON/KOMLU2/KOMLU2(11) COMMON POINTERS/FLAGS FOR UNIT 2
COMMON/KOMLU3/KOMLU3(11) COMMON POINTERS/FLAGS FOR UNIT 3
COMMON/KOMLU5/KOMLU5(11) COMMON POINTERS/FLAGS/BUFFER FOR UNIT 5
COMMON/KOMLDN/KOMLDN(11) COMMON I/O PACKETS FOR DETECTION FILES (21-24)
COMMON/KOMNRM/KOMNRM(11) COMMON ERTS SCENE PARAMETERS
COMMON/KOMNET/KOMNET(11) COMMON CONTROL NETWORK COORDINATES
COMMON/KOMLWM/KOMLWM(11) COMMON OUTPUT WINDOW PACKETS
COMMON/KOMSLM/KOMSLM(11) COMMON SPECTRAL LIMITS
COMMON/KOMSYM/KOMSYM(11) COMMON SYMBOL TABLE
COMMON/KOMTBL/KOMTBL(11) COMMON MULTI-PURPOSE TABLE
INCLUDE KOMXT.List COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULLCST.List DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

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 KOTYP 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 POKE TRUE IF POKE CALLED. FALSE IF NOT

PROCEDURE

CALL TRACE

GET NAME OF COMMON

CALL MCIOULN( NULCST)
NAMCOM=???
CALL GET5KIH(NAMCOM,11, NULCST)

GET/CHECK LOCATION OF VARIABLE TO 'POKE' AT

CALL GET5IN(LOCPOK, 1.2500. 'BAD POKE LOCATION --')

GET/DECODE VALUE TO INSERT

KHPOKE(1)=??
CALL GET5KIH(KHPOKE,(12). NULCST)
CALL DCOE(INPOKE,RLPOKE,KOTYP)

M-87
KMPK0K: 1(1) (LENCST(KMPK0K.18)))
IF(KMPK0K (1) .EQ. '??? ' ) CALL WARNIN('BAD VALUE TO INSERT --')
CALL GETSIN(LOCPOK. 11-.1,'EXTRA POKE SPECIFICATION --')

GIVE UP IF ANY WARNINGS OCCURRED
IF (NDWARN.NE.0) GO TO 900

FIND REQUESTED COMMON & 'POKE' AT IT
POKE=.FALSE.
IF (NAMCOM.EQ. 'KOMALT') CALL POKE(KOMALT.KOMALT)
IF (NAMCOM.EQ. 'KOMDET') CALL POKE(KOMDET.KOMDET)
IF (NAMCOM.EQ. 'KOMFIT') CALL POKE(KOMFIT.KOMFIT)
IF (NAMCOM.EQ. 'KOMH RT') CALL POKE(KOMHRT.KOMHRT)
IF (NAMCOM.EQ. 'KOMIWH') CALL POKE(KOMIWH.KOMIWH)
IF (NAMCOM.EQ. 'KOMKLS') CALL POKE(KOMKLS.KOMKLS)
IF (NAMCOM.EQ. 'KOMLOG') CALL POKE(KOMLOG.KOMLOG)
IF (NAMCOM.EQ. 'KOMSLM') CALL POKE(KOMSLM.KOMSLM)
IF (NAMCOM.EQ. 'KOMSYM') CALL POKE(KOMSYM.KOMSYM)
IF (NAMCOM.EQ. 'KOMTBL') CALL POKE(KOMTBL.KOMTBL)
IF (NAMCOM.EQ. 'KOMXQT') CALL POKE(KOMXQT.KOMXQT)
IF (.NOT.POEKE) CALL MOWARN('BAD POKE COMMON NAME -- --'.CBS4CS(NAMCOM,(1),(8)))

DONE
900 KOND=" 
CALL NOCLAW(NULCST)
RETURN

INTERNAL
SUBROUTINE POKE( & PRINT & CHANGE & PRINT VALUE OF COMMON LOCATION
I KOM, & LABELLED COMMON (INTEGER OR CHARACTER)
I COM, & LABELLED COMMON (REAL)
C)
INTEGER KOM(I) & ARGUMENT
REAL COM(I) & ARGUMENT
REAL RTEMP & TEMPORARY
INTEGER NFMT & TYPE OF 'F' FORMAT TO PRINT RTEMP WITH
INTEGER LOCHI & LOCATION OF FIRST QUOTE
INTEGER LOCH2 & LOCATION OF SECOND QUOTE
C)
POKED=.TRUE.
RLTEMP=CM(LOCPOK)+ZERO
0 FORCE NORMALIZATION IF NOT REAL
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,
6 161) NAMCOM.LOCPK.KOM(LOCPOK).RLTEMP.KOM(LOCPOK)
161 FORMAT(1X.A6,' (.J4.'1 ,I,0.TX.F4.0.A6)
IF(NFORMT.EQ.2) WRITE(6,
6 162) NAMCOM.LOCPK.KOM(LOCPOK).RLTEMP.KOM(LOCPOK)
162 FORMAT(1X.A6,' (.J4.'1 ,I,0.TX.F4.0.A6)
IF(NFORMT.EQ.3) WRITE(6,
6 163) NAMCOM.LOCPK.KOM(LOCPOK).RLTEMP.KOM(LOCPOK)
163 FORMAT(1X.A6,' (.J4.'1 ,I,0.TX.F4.0.A6)
WRITE(6,169)
169 FORMAT('CHANGED TO')
IF(KOOTYP.EQ.'IN') KOM(LOCPOK)=INPOKE
IF(KOOTYP.EQ.'RL' .OR.
6 KOOTYP.EQ.'FR' .OR.
6 KOOTYP.EQ.'SX') COM(LOCPOK)=RLPOKE
IF(KOOTYP.EQ.'ERR') GO TO 200
LOCCH = LCHREQ(INPOKE(11).1.6.*)
IF(LCHCH.EQ.0) GO TO 190
LOCCH = LCHREQ(RLPOKE(11).LOCHI+1.6-LOCCH.*)
IF(LCHCH.EQ.0) LOCCH = 7
CALL MOVCS(KHPOKE(11).1.6.
6 KHPOKE(11).LOCHI+1.LOCCH+(LOCCH+1).*)
190 KOM(LOCPOK) = KHPOKE(11)
200 RLTEMP=CM(LOCPOK)+ZERO
0 FORCE NORMALIZATION IF NOT REAL
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,
6 161) NAMCOM.LOCPK.KOM(LOCPOK).RLTEMP.KOM(LOCPOK)
161 FORMAT(1X.A6,' (.J4.'1 ,I,0.TX.F4.0.A6)
IF(NFORMT.EQ.2) WRITE(6,
6 162) NAMCOM.LOCPK.KOM(LOCPOK).RLTEMP.KOM(LOCPOK)
162 FORMAT(1X.A6,' (.J4.'1 ,I,0.TX.F4.0.A6)
IF(NFORMT.EQ.3) WRITE(6,
6 163) NAMCOM.LOCPK.KOM(LOCPOK).RLTEMP.KOM(LOCPOK)
RETURN
END
SUBROUTINE KNDPOL ( & GET/CHECK POLAR GAIN/BIAS  
 U KOMB) 8 1: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY
---
E H SCHLOSSER LEC 12/29/78 ORIGINAL CODE
E H SCHLOSSER LEC 07/30/78 DELETES RETURN K
E H SCHLOSSER LEC 08/03/78 REVISE GETS... CALLS
E H SCHLOSSER LEC 12/09/78 UPGRADE DOCUMENTATION

METHOD
---
UPDATE POLAR GAIN(S)/BIAS(ES) FROM UNIT 5. IF SPECIFIED, AND CONFIRM.  
(BIASES ARE MAINTAINED IN LABELLED COMMON AS INTEGERS SCALED BY 2**-4.)  
(GAINS ARE MAINTAINED IN LABELLED COMMON AS INTEGERS SCALED BY 2**12.)

MACHINE-DEPENDENT CODE
------------------
SCALE INTEGER BIASES & GAINS ASSUME 36-BIT INTEGERS.  
DIMENSION & FORMAT SPECIFICATIONS ASSUME 6 CHARACTERS PER INTEGER.

EXTERNAL REFERENCES
---------------------
GETSIN 3 GET INTEGER DATA FIELD FROM UNIT 5
GETSCH 3 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSRL 3 GET REAL DATA FIELD FROM UNIT 5
WARNS 3 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5

EXCEPTIONS
----------
1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE  
   'WARNING' DIAGNOSTICS:
   COEFFICIENT TYPE <= 'GAI' OR 'BIA'
   1 <= CHANNEL NUMBER <= 2
   -128. <= GAIN <= +128.
   -100. <= BIAS <= +100.

GLOBAL DECLARATIONS
---------------------
INCLUDE KOMKOT.LIST 3 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMIRL.LIST 3 COMMON IRRADIANCE TRANSFORM TYPE & COEFFICIENTS
INCLUDE NULCST.LIST 3 DEFINE NULL CHARACTER STRING

M-90
C LOCAL DECLARATIONS
C

INTEGER KTEMP         3 TEMPORARY
INTEGER NPCHAN        8 POLAR CHANNEL NUMBER
REAL PA(2),PB(2)      8 TEMP STORAGE FOR GAINS/BIASES AS RL NUMBERS
INTEGER NAME(2) /'RADIUS', 'ANGLE'/ 8 INTUITIVE NAMES FOR POLAR.1 & 2

C PROCEDURE
C

CALL TRACE

C RETRIEVE CURRENT POLAR GAINS/BIASES

PA(1)=FLOAT(NR012(1))/2.**12
PA(2)=FLOAT(NR012(2))/2.**12
PB(1)=FLOAT(NR084(1))/2.**24
PB(2)=FLOAT(NR084(2))/2.**24

C GET/CHECK POLAR TRANSFORMED CHANNEL NUMBER

NPC=0
CALL GETSIN(NPC, 1.2,'BAD POLAR CHANNEL NUMBER - - -')
IF(NPC.EQ.0) GO TO 890 8 NONE OR INVALID

C GET TRANSFORMATION COEFFICIENT TYPE

300 KTEMP=' NUL'
CALL GETSKH(KTEMP,(3), NULCST)

C GET/CHECK/CONFIRM GAIN

400 IF(KTEMP.NE.' GAIN') GO TO 500
CALL GETSLR(PA(NPC), NRT012(NPC)+PA(NPC tumble)+2.**12
IF(HCFIRM.EQ.0) WRITE(6,425) NPC, NAME, PA(NPC)
425 FORMAT(' POLAR. ','GAIN', A, 9.3)
GO TO 300

C GET/CHECK/CONFIRM BIAS

500 IF(KTEMP.NE.' BIAS') GO TO 800
CALL GETSLR(PB(NPC), NRT084(NPC)+PB(NPC tumble)+2.**24
IF(HCFIRM.EQ.0) WRITE(6,525) NPC, NAME, PB(NPC)
525 FORMAT(' POLAR. ','BIAS', A, 9.3)
GO TO 300

C
C: FLAG BAD SPECIFICATION
C
800 IF(KHTEMP.EQ. 'NUL') GO TO 900
   CALL H HệM ( 'BAD POLAR SPECIFICATION --')
C
C: CONFIRM GAINS AND BIASES
C
890 IF(MCFIRM.EQ.0) GO TO 900
DO 880 NPCHAN=1,2
   WRITE(6,425) NPCHAN,NANE(NPCtAN),POAI(NPCtAN)
   WRl'TE(6.525) NPCHAfN,NANE(NPCtAN),PBIAS(NPCtAN)
880 CONTINUE
C
C: NORMAL RETURN
C
900 KOMO=' ' RETURN
END
SUBROUTINE KMDPRI: 8 GET/CHECK PRINTER SPECIFICATIONS
U KOND: 8 1: FIRST 3 CHAR OF COMMAND 0: SPACES

HISTORY

E H SCHLOSSER LEC 01/19/78 ORIGINAL CODE
E H SCHLOSSER LEC 07/30/78 DELETE RETURN K
E H SCHLOSSER LEC 02/29/79 REVISE GET.. CALLS
E H SCHLOSSER LEC 12/05/79 UPGRADE DOCUMENTATION

METHOD

UPDATE PRINTER CHARACTERISTICS FROM UNIT 5. IF SPECIFIED. AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

MDWARN 8 PRINT/COUNT/LOG "WARNING" DIAGNOSTIC MESSAGE
GETSKH 8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
WARNS 8 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
GETSIN 8 GET INTEGER DATA FIELD FROM UNIT 5

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   THE ALTERNATE PRINT FILE(S) ARE ALREADY OPEN (I.E.: THE FIRST WINDOW HAS ALREADY BEEN PROCESSED).

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE 'WARNING' DIAGNOSTICS:
   PRT FUNCTION = 'INC' OR 'PAO' OR 'DEV' OR 'FIL' OR 'CON'
   2 <= LINES/INCH < 20
   2 <= COLUMNS/INCH < 20
   40 <= LINES/PAGE < 99
   40 <= COLUMNS/PAGE < 132 8 NOTE: BUT UPDATED IF <= 1000
   0 <= NUMBER FILES <= MIND(RHLTH..RSLTH)
   DEV MNEMONIC = (ANYTHING OTHER THAN ALL BLANKS)
   PRINT CONTROL = 'AUT' OR 'MAN' OR 'NON'

3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC MESSAGE.
GLOBAL DECLARATIONS

INCLUDE \$NAM.FLT  
& COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST  & DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

INTEGER NAMFUN  & NAME OF PRINTER FUNCTION BEING SPECIFIED

PROCEDURE

CALL TRACE

CHECK IF COMMAND IS NOW LEGAL

IF(NAMF<0<.LT.-1) GO TO 100
  CALL WARNING:
  'PRINTER COMMAND CANNOT BE USED AFTER FIRST WINDOW IS PROCESSED'
  GO TO 900

GET/CHECK PRINTER FUNCTION TO BE SPECIFIED

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

FLAG BAD FUNCTION & DRAIN SPECS

CALL WARNING:
  'BAD PRINTER FUNCTION --'
150 NAMFUN• 'NUL'
  CALL GETSKH(NAMFUN.(3), NULCST)
  IF(NAMFUN.NE.'NUL') GO TO 150

GET/CHECK/CONFIRM LINES & COLUMNS PER INCH

200 CALL GETSIN(LINCN.  2.80.'BAD LINES/INCH --'
  CALL GETSIN(KINCN.  2.80.'BAD COLUMNS/INCH --'
  CALL GETSIN(NAMFUN. +1.-1.'EXTRA PRINTER SPEC --'
  YLINCH=LINCN
  XLINCH=KINCN
  IF(NAMFUN.EQ.'NUL') WRITE(6,215) LINCN,KINCN
215 FORMAT('PRINTER. INCH. '.13.' LINES. '.13.' COLUMNS')

H-94
DAN PACKAGE APPENDIX M
COMMAND Routines

IF(NAMFUN.NE.' NUL') GO TO 900

C GET/CHECK/CONFIRM LINES & COLUMNS PER PAGE

300 CALL OETSKN IMEMONIC
CALL OETSKM(MEMONIC(8), NULCST)
C GET/CHECK/CONFIRM DEVICE TYPE MNEMONIC

CALL OETSKMINAMITN
CALL OETSKMINAMITN, 0, NAMITN
C GET/CHECK/CONFIRM NUMBER OF FILES

900 MALTM=MALTM(MALTM, MALTM)
CALL OETSKMINALTM, 0, MALTM
C GET/CHECK/CONFIRM PRINTER CONTROL

800 KONPR=KONPR
CALL OETSKMINKONPR(3), NULCST)
IF(KONPR.EQ.' AUT') KONAME=' AUTO'
IF(KONPR.EQ.' MAN') KONAME=' MAN'
IF(KONPR.EQ.' NON') KONAME= ' NUL'
IF(KONPR.EQ.' BAD') CALL WARNS(' BAD CONTROL SPEC ---')
IF(KONPR.NE.0) WRITE(6,819) KONAME
C
819 FORMAT(' PRINTER. CONTROL. 'AB)

900 KONAM='
RETURN
END
SUBROUTINE KMDRAD: 0 GET/CHECK RADIANCE LIMITS
U KMDRD: 1: FIRST 3 CHAR OF COMMAND 0: SPACES
---

HISTORY
-----
E H SCHLOSSER LEC 09/13/73 NUMERIC OPTION
E H SCHLOSSER LEC 12/17/79 ALPHANUMERIC COMMAND
E H SCHLOSSER LEC 07/17/79 DELETE RETURN K
E H SCHLOSSER LEC 02/03/79 REVISE GETS... CALLS
E H SCHLOSSER LEC 12/11/79 UPGRADE DOCUMENTATION

METHOD
------
UPDATE RADIANCE LIMITS FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE
-------------------------
NONE.

EXTERNAL REFERENCES
---------------------
GETSKN 0 GET CHARACTER STRING DATA FIELD FROM UNIT 9
UNOGFN 0 BACK UP ONE DATA FIELD ON UNIT 9
GETBIN 0 GET INTEGER DATA FIELD FROM UNIT 5
HONAPN 0 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE

EXCEPTIONS
-----------
1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE 'WARNING' DIAGNOSTICS:
   0 <= LOW RADIANCE <= 127
   LOW RADIANCE <= HIGH RADIANCE <= 127
2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS
---------------------
INCLUDE KOMKOT.LIST 0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST 0 COMMON CLASSIFICATION INFO
INCLUDE NULCST.LIST 0 DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS
C

GAM PACKAGE APPENDIX M

CONRAI)D ROUTINE'S

----------

C

PROCEDURE

----------

C

GET/CHECK RADIANCE SPECIFICATIONS FOR LIMIT CHANNEL(S)

C

CONFIRM RADIANCE LIMITS

C

NORMAL RETURN

C

900

RETURN

END
(NOT IMPLEMENTED)
SUBROUTINE KMDREN: 0 GET/CHECK NEW WINDOW SEQUENCE NUMBER
U KOKO) 0 1: FIRST 3 CHARs OF COMMAND 0: SPACES

HISTORY

E H SCHLOSSER LEC 10/20/75 ORIGINAL CODE
E H SCHLOSSER LEC 07/17/78 DELETE RET K
M A TOMPKINS LEMSCO 09/27/80 UPGRADE DOCUMENTATION

METHOD

IF NWNDOW <> INITIAL STATE THEN GET NEW WINDOW NUMBER FROM UNIT 5.
IF SPECIFIED. AND CONFIRM. INITIALIZE NWNDOW TO NEW VALUE
MAINTAINING PRESENT STATE.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSIN 3 GET INTEGER DATA FIELD FROM UNIT 5
WARNS 3 PROCESS WARNING DIAGNOSTIC FOR UNIT 5

EXCEPTIONS

1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   1 <= WINDOW SEQUENCE NUMBER <= 999
2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC MESSAGE.

GLOBAL DECLARATIONS

INCLUDE KOMXOT.LIST 3 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS

LOCAL DECLARATIONS

INTEGER ITEmP 3 ABSOLUTE OF CURRENT WINDOW NUMBER
C PROCEDURE
C *******
C
CALL TRACEN
C
IF(NWINDO.EQ.0) GO TO 800
I TEMP=ABS(NWINDO)
CALL GETSIN(ITEMP, 1.999, 'BAD WINDOW NUMBER --')
IF(NCFIRM.NE.0) WRITE(6,125) ITEMP
125 FORMAT(' RENUMBER. ',I3)
C
ALTER NWINDO TO ITE MP KEEPING ORIGINAL SIGN (STATE).
C
NWINDO=SIGN(ITEMP,NWINDO)
GO TO 900
800 CALL WARN5( 'INVALID DEFAULT COMMAND --')
900 KOND=' '
RETURN
END
DAN PACKAGE APPENDIX M
COMMAND ROUTINES

(NOT IMPLEMENTED)
SUBROUTINE KNDSCCA: 8 GET/CHECK WINDOW SCALE
U KOND) 8 1: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY
--------

E H SCHLOSSER 08/05/73 NUMERIC OPTION
E H SCHLOSSER LEC 12/07/75 ALPHANUMERIC COMMAND
E H SCHLOSSER LEC 07/17/78 DELETE RETURN K
E H SCHLOSSER LEC 02/20/79 GETS.. CALLS
M A TOMPKINS LEMSCO 09/27/80 UPGRADE DOCUMENTATION

METHOD
------

UPDATE WINDOW SCALE FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

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

NONE.

EXTERNAL REFERENCES
---------------------

OETSIN 8 GET INTEGER DATA FIELD FROM UNIT 5
OETSFR 8 GET FRACTION DATA FIELD FROM UNIT 5

EXCEPTIONS
-----------

1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   1.000000. < SCALE <= 1.000000.
   2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS
---------------------

INCLUDE KOMXOT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMFIT.LIST 8 ADJUSTMENT/REGISTRATION PARAMETERS

LOCAL DECLARATIONS
--------------------

INTEGER IRFABS 8 ABSOLUTE OF REPRESENTATIVE FRACTION DENOMINATOR
REAL FRTEMP 8 TEMPORARY FRACTION

M-102
PROCEDURE

CALL TRACE

IRFABS = \|ABS(IRFD)
FRTEMP = 1.0/\(\text{FLOAT} (\|RFABS) + 0.00001\)
CALL GET5FR(FRTEMP, 1.0/1000000.0, 1.0/20000.0, 'BAD SCALE ---')
IRFABS = (1.0/FRTEMP) + 0.5

100 CALL GET5IN(DUMMY, +1.0, 'EXTRA SCALE SPECIFICATION ---')
IF(HCFIRM.EQ.0) GO TO 900
IF(IRFABSGT.99999) GO TO 140
WRITE(8,125) IRFABS
125 FORMAT(' SCALE, /', F15.15)
GO TO 900
140 WRITE(8,145) IRFABS
145 FORMAT(' SCALE, /', F15.15)

IF SCALE WAS CHANGED, FLAG AS UNCALIBRATED (NEGATIVE)

900 IF(\|ABS(IRFD) .NE. \|RFABS) IRFD = -\|RFABS
RETURN
END
SUBROUTINE KMOSCE  & GET/CHECK ERTS SCENE NUMBER
U1 KOMDI & 1: FIRST 3 CHAR'S OF COMMAND 0: SPACES

HISTORY

E M SCHLOSSER LEC 12/08/79 ALPHANUMERIC COMMAND
E M SCHLOSSER LEC 07/18/78 DELETE RETURN K
E M SCHLOSSER LEC 02/18/79 REVISE DOCUMENTATION & OETS... CALLS
E M SCHLOSSER LEC 01/06/80 MAKE SAMPLES/SCENE OPTIONAL

METHOD

UPDATE SCENE NUMBER FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSKN  8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN  8 GET INTEGER DATA FIELD FROM UNIT 5
MDWARN  8 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE

EXCEPTIONS

1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   1 <= LSAT <= 6
2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KMONXT.LIST  8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KMONER.LIST  8 COMMON ERTS SCENE PARAMETERS
INCLUDE NULCST.LIST   8 DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

INTEGER JSCENE(2)
PROCEDURE

CALL TRACE

GET SCENE NUMBER

JSCENE(1) = ' NUL'
CALL GETSK(JSCENE, (1), NULCST)
IF(JSCENE(1).EQ. ' NUL') Go TO 300

DECODE PRE-JAN79 SCENE NUMBER (3 DIGITS FOR DAYS SINCE LAUNCH)

DECOD(10, 115, JSCENE, ERR=140) NERTS
115 FORMAT(11, 13, 16)
NERTS(3) = NERTS(3)
Go TO 200

DECODE POST-JAN78 SCENE NUMBER (4 DIGITS FOR DAYS SINCE LAUNCH)

140 DECOD(11, 145, JSCENE, ERR=180) NERTS
145 FORMAT(11, 14, 16)
NERTS(3) = NERTS(3)
Go TO 200

180 CALL WARN5 ( 'BAD SCENE NUMBER --')

GET OPTIONAL SAMPLES/SCENE (TEMPORARY FOR COMPATIBILITY WITH PRE- 7912 SYNTAX)

INTEMP = -9999
CALL GETSIN(INTEMP, 700, 7000, 'BAD SAMPLES/SCENE --')
IF(INTEMP.NE. -9999) NERSAM = INTEMP
IF(NERSAM.LT. 900) NERSAM = 900
IF(INTEMP.NE. -9999) CALL MONOTE( 'CSAM1(NERSAM, 5), 'SAMPLES PER SCENE')
CALL GETSIN(IDUMMY, 1, -1, 'EXTRA SCENE SPECIFICATION --')

CONFIRM SCENE NUMBER

300 IF(MCFIRM .NE. 0) WRITE(6, 255) NERTS
255 FORMAT(' SCENE: ', 11, JW, ' ', J5)
IF(NERTS(3).LT.0) CALL MDWARN1 ('ADJUSTMENT WILL NOT INCLUDE NON-LINEAR CORRECTIONS')

NORMAL RETURN

KOMO = '
RETURN

M-105
DAM PACKAGE APPENDIX M
COMMAND ROUTINES

END
(NOT IMPLEMENTED)
SUBROUTINE KROSHA: 8 GET/CHECK SHARPENING FILTER COEFFICIENTS
U KONDO: 8 1: FIRST 3 CHAR OF COMMAND 0: SPACES

--- HISTORY ---
E H SCHLOSSER LEC 09/18/78 ORIGINAL CODE
M A TOMPKINS LEMSCO 09/27/80 UPGRADE DOCUMENTATION

--- METHOD ---
CHECK FOR VALID CHANNEL AND VERIFY THAT THE INPUT AXIS IS
A SAMPLE AXIS. UPDATE SHARPENING FILTER COEFFICIENTS FROM UNIT
5, IF SPECIFIED, AND CONFIRM.

--- MACHINE-DEPENDENT CODE ---
ALIGNMENT OF SCALED INTEGERS ASSUMES 32-BIT COMPUTER WORD.

--- EXTERNAL REFERENCES ---
GETBIN 8 GET INTEGER DATA FIELD FROM UNIT 5
GETSKN 8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSR 8 GET REAL DATA FIELD FROM UNIT 5
WARN 8 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5

--- EXCEPTIONS ---
1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
 'WARNING' DIAGNOSTICS:
 -0.1 <= 1ST COEFFICIENT <= 0.1
 -0.8 <= 2ND COEFFICIENT <= 0.8
 1 <= CHANNEL NUMBER <= NERSHA

2. ANY AXIS INPUT TO BE SHARPENED OTHER THAN A SAMPLE AXIS
 GENERATES A 'WARNING' DIAGNOSTIC.

--- GLOBAL DECLARATIONS ---
INCLUDE KOMXOT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONNER.LIST 8 COMMON ERS SCENE PARAMETERS
INCLUDE KOMINT.LIST 8 COMMON IRRADIANCE TRANSFORM TYPE & COEFFICIENTS
INCLUDE NULLCST.LIST 8 DEFINE NULL CHARACTER STRING
C LOCAL DECLARATIONS
C ---------------
C
INTEGER NSCH 8 NUMBER SHARPENING CHANNEL
INTEGER NCOE 8 NUMBER OF COEFFICIENTS
INTEGER NSCHAN 8 CHANNEL TO BE SHARPENED
REAL SFCOEF(9,8) 8 TEMP STORAGE FOR COEFFICIENTS AS REAL NUMBERS

C PROCEDURE
C ------------
C
CALL TRACE
C
C RETRIEVE CURRENT COEFFICIENTS
C
DO 220 NSCH=1,NERCHA
DO 210 NCOE=1,2
210 SFCOEF(NSCH,NCOE)=FLOAT(IRSFL(NSCH,NCOE))/2.**12
220 CONTINUE
C
C GET/CHECK CHANNEL NUMBER TO BE SHARPENED
C
NSCHAN='NONE'
CALL OETSIN(NSCHAN,1,NERCHA,'BAD CHANNEL NUMBER --')
IF(NSCHAN.EQ.'NONE') GO TO 850 8 NONE OR INVALID
C
C GET/CHECK AXIS TO BE SHARPENED
C
ITEMP='NONE'
CALL OETSKHITEMP,(3),NULCST)
IF(ITEMP.EQ.'SAM') GO TO 400 8 ONLY SHARPENING OF SAMPLE AXIS SUPPORTED
CALL WARNH('BAD SHARPENING AXIS --')
GO TO 850
C
C GET/CHECK/CONFIRM COEFFICIENTS
C
400 CALL OETSR(SFCOEF(NSCHAN,1),1,-0.1,-0.1,'BAD COEFFICIENT --')
IRSPFL(NSCHAN,1)=SFCOEF(NSCHAN,1)*2.**12
CALL OETSR(SFCOEF(NSCHAN,2),1,-0.8,-0.8,'BAD COEFFICIENT --')
IRSPFL(NSCHAN,2)=SFCOEF(NSCHAN,2)*2.**12
IF(INCFIRM.NE.0) WRITE(6,925) NSCHAN,(SFCOEF(NSCHAN,NCOE),NCOE=1,2)
925 FORMAT('SHARPENING','N', SAMPLE,'F7.4','F7.4')
GO TO 800
C
C CONFIRM ALL SHARPENING FILTER COEFFICIENTS
C
850 IF(INCFIRM.EQ.0) GO TO 900
DO 860 NSCHAN=1,NERCHA
WRITE(6,925) NSCHAN,(SFCOEF(NSCHAN,NCOE),NCOE=1,2)
860 CONTINUE
899
850
DAN PACKAGE APPENDIX H
COMMAND ROUTINES

080 CONTINUE
C
C NORMAL RETURN
C
900 KDMD0
    RETURN
    END

KRDMA 093
SUBROUTINE KNDSIZ1: GET/CHECK SCENE SIZE IN SCAN COORDINATES

U RONDI: 0:1: FIRST 3 CHAR OF COMMAND J: SPACES

HISTORY

E N SCHLOSSER LEC 12/19/78 RMTS/DESIGN/CODE

METHOD

UPDATE SCENE SIZE FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSCHM 8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN 8 GET INTEGER DATA FIELD FROM UNIT 5
WARN8 8 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
MHWARM 8 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   SCANN SPECIFICATION MISSING OR MISSPELLED

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   2000 <= SCAN LINES <= 4000
   2000 <= SCAN SAMPLES <= 4000

3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KONKOT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KONKNER.LIST 8 COMMON CRTS SCENE PARAMETERS
INCLUDE NULCST.LIST 8 DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS

M-111
INTEGER NOSAVE  // SAVE AREA FOR CONTENTS OF NDTOTL ON ENTRY
INTEGER KHTEMP  // A TEMPORARY

C
C
C"

CALL TRACE
C
C GET/CHECK SCAN (OSFC-ADJUSTED) COORDINATES
C
NOSAVE=NDTOTL
KHTEMP='NUL'
CALL GETSKH(KHTEMP,(3), NULCST)
IF(KHTEMP.EQ.'NUL') GO TO 400  // NO SPECS -- CONFIRM
IF(KHTEMP.EQ.'SCA') CALL WARNS('COORDINATE SYSTEM NOT SCAN --')
IF(KHTEMP.NE.'SCA') GO TO 400
CALL GETSKN(KHTEMP,'NULCST',
CALL GETSKS(NERLIN, 2000.0000,'BAD LINES --')
CALL GETSKS(NERSAM, 2000.4000,'BAD SAMPLES --')
CALL GETSKS(KHTEMP, +1,-1,'EXTRA SIZE SPECIFICATION --')

C
C CONFIRM SIZE COORDINATES
C
400 IF(MCFIRM.NE.0) WRITE(8,44S1) NERLIN,NERSAM
44S FORMAT(' SIZE. SCAN. ',15.,',',15)
C
C
C NORMAL RETURN
C
900 KOMD='-'
RETURN
END
SUBROUTINE KMSPA! A GET/CHECK/INSERT WINDOW SPACING
U.KOMDI A 1: FIRST 3 CHAR OF COMMAND 0: SPACES

METHOD

GET THE LINE AND SAMPLE SPACING AS REAL VALUES.
MULTIPLY BY 100. AND CONVERT TO INTEGERS AND STORE
IN THE SCANNER OUTPUT WINDOW. CONFIRM IF ON.

EXTERNAL REFERENCES

GETSRFR A GET INTEGER/REAL/FRACTION FROM UNIT S
GETSIN A GET INTEGER DATA FIELD FROM UNIT S

EXCEPTIONS

1. ANY SPACING SPECIFICATION LESS THAN 0.2 OR GREATER
   THAN 20 OR NOT NUMERIC GENERATES A WARNING AND DOES
   NOT CHANGE THE CURRENT SPACING.
2. ANY EXTRA SPECIFICATION GENERATES A WARNING.

GLOBAL DECLARATIONS

INCLUDE KONXOT.LIST A COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMOWW.LIST A COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDIF.LIST 8 DEFINE STRUCTURE OF WINDOW PACKETS

LOCAL DECLARATIONS

M-113
REAL SPL100  0 LINE SPACING * 100  
REAL SPS100  0 SAMPLE SPACING * 100  
INTEGER ILINSP  0 LINE SPACING  
INTEGER ISAMSP  0 SAMPLE SPACING  

C
C PROCEDURE
C
C CALL TRACE
C
C GET LINE AND SAMPLE SPACING SPECS
C
SPL100=MSAOWN(MLIN.WSP100)
SPS100=MSAOWN(WSAM.WSP100)
CALL OETSFR(SPL100.
  100..1999.2000.01. 'BAD LNI SPACING --')
CALL OETSFR(SPS100.
  100..1999.2000.01. 'BAD SAMPLE SPACING --')
MSAOWN(MLIN.WSP100)=SPL100+0.5
MSAOWN(WSAM.WSP100)=SPS100+0.5
KSYOWN(WSP100)='SCA'

C
C FLUSH REMAINING SPECS
C
CALL OETSIN(IDUMMY.
  '-'-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(6,800) ILINSP,ISAMSP
800 FORMAT(' SPACING. ',-12,' LINES. '12,' SAMPLES')
C
C CLEAR THE COMMAND
C
DAM PACKAGE APPENDIX M
COMMAND ROUTINES

900 KOND="
C
C
C RETURN TO CALLING ROUTINE
C
900 RETURN
END
(NOT IMPLEMENTED)
SUBROUTINE KMDSYN! 0 GET/CHECK WINDOW SYMBOLS
U KOND) 0 I: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY
-------
E H SCHLOSSER LEC 09/21/73 NUMERIC OPTION
E H SCHLOSSER LEC 12/03/75 ALPHANUMERIC COMMAND
E H SCHLOSSER LEC 07/18/78 DELETE RETURN K
E H SCHLOSSER LEC 02/21/79 REVISE GETS.. CALLS
E H SCHLOSSER LEC 11/28/79 REMOVE UNIVAC FLD FUNCTION

METHOD
------
GET, CHECK, INTERPOLATE IF NECESSARY, AND STORE THE
SYMBOL CHARACTER STRINGS FOR LINE-PRINTER MAPS IN CHARACTERS 1
THRU 4 OF THE KSYM ARRAY ELEMENTS.

MACHINE-DEPENDENT CODE
----------------------
NONE.

EXTERNAL REFERENCES
--------------------
GETSKH 0 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSIN 0 GET INTEGER DATA FIELD FROM UNIT 5
WARNS 0 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
UNGETS 0 BACK UP ONE DATA FIELD ON UNIT 5
PMWARN 0 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
MOVCST 0 MOVE CHARACTER STRING
INTEGER LENCST 0 LENGTH OF CHARACTER STRING
INTEGER ICE 0 INTEGER CHARACTER EQUIVALENT

EXCEPTIONS
------------
1. MAXIMUM NUMBER FOR WHICH A SYMBOL MAY BE DEFINED IS ISYMH1.
2. SYMBOLS LONGER THAN 4 CHARACTERS ARE TRUNCATED TO 4 CHARACTERS.
3. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS
----------------------
INCLUDE KOMXQT.LIST 0 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMSYM.LIST  # COMMON SYMBOL TABLE
INCLUDE NULCST.LIST  # DEFINE NULL CHARACTER STRING
INCLUDE MAXINT.LIST  # DEFINE MAXIMUM INTEGER

C LOCAL DECLARATIONS

C------------------
C
INTEGER NOSAVE  # TEMPORARY SAVE FOR CONTENTS OF NOTOTL ON ENTRY
INTEGER MINSYM,MAXSYM  # MINIMUM, MAXIMUM SYMBOL
INTEGER LENSYM  # LENGTH OF SYMBOL
INTEGER NUM  # NUMBER (RAD/GRAM/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 SYINUM  # SYMBOLS PER NUMBER (FOR INTERPOLATION)

C PROCEDURE

-----

CALL TRACE

C INITIALIZE

NOSAVE=NOTOTL

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'
6 (MINSYM.EQ.'**') OR.  # RESERVED FOR PRIMARY TICKS
6 (MINSYM.EQ.'**') OR.  # RESERVED FOR SECONDARY TICKS
6 CALL WARNH('BAD FIRST SYMBOL --')

GET REQUIRED MINIMUM NUMBER (INTEGER) FROM 2ND SPEC FIELD

MINNUM=MAXINT
CALL GETSN(MINNUM,0.1SYMHI.'BAD FIRST NUMBER --')

C INITIALIZE IMPLICIT MAXIMUM SYMBOL AND NUMBER

MAXSYM=MINSYM
MAXNUM=MINNUM

C GET THIRD SPEC FIELD AS BOTH CHARACTER STRING & NUMBER

KH3SYM='NUL'
IN3NUM=-999
CALL GETSKM(KH3SYM, 4, NULCST)
CALL UNGETS
CALL GETSIN(IN3NUM, 0, ISYMH, NULCST)

C GET FOURTH SPEC FIELD AS BOTH CHARACTER STRING & NUMBER
C
KH4SYM=' NUL'
IN4NUM=-999
CALL GETSKM(KH4SYM, 4, NULCST)
CALL UNGETS
CALL GETSIN(IN4NUM, 0, ISYMH, NULCST)

C ASSIGN EXPLICIT MAXIMUM SYMBOL AND NUMBER
C
IF(KH4SYM.EQ. ' NUL') GO TO 120
MAXSYM=KH4SYM
MAXNUM=IN4NUM
GO TO 140
120 IF(KH4SYM.NE. ' NUL') MAXNUM=IN3NUM
140 CONTINUE

C CHECK MAXIMUM SYMBOL AND NUMBER
C
IF((MAXSYM.EQ. '.').OR. 8 RESERVED FOR 'NO DATA'
& (MAXSYM.EQ. '+').OR. 8 RESERVED FOR PRIMARY TICKS
& (MAXSYM.EQ. '=').OR. 8 RESERVED FOR SECONDARY TICKS
& CALL MDWARN( 'BAD SECOND SYMBOL')
IF(MAXNUM.LT.0) CALL MDWARN( 'BAD SECOND NUMBER')

C COMPUTE SYMBOLS PER NUMBER
C
MAXNUM=M1NO(MAXNUM, ISYMH)
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 CHECK FOR EXTRA SPEC FIELDS & FOR DIAGNOSTICS
C
CALL GETSIN(IN4NUM, +1,1,'EXTRA SYMBOL SPECIFICATION ---')
IF(NOSAVE.NE.NDTOTL) GO TO 900

C LOAD SYMBOLS (CHARACTER STRINGS) INTO SYMBOL PART (CHARS 1-4) OF SYMBOL TABLE
C
ROUND=SIGN(IN4SYM,1-E-B.SYINUM)
DO 300 NUM=M1NO(MAXNUM,MAXNUM)
   CALL PUTICE(KSYM(NUM-11,1, (1),
                 ICE(MINSYM)+IFIX(FLOAT(NUM-MINNUM)*SYINUM-ROUND))
   CALL MOCVST(KSYM(NUM-11,2,3),
                 MINSYM,2,3,' ')& REPLICATE NEXT 3 CHAR
C
C
C
M-119
300 CONTINUE

C
C CONFIRM SYMBOLS
C
IF(MCFIRM.EQ.0) GO TO 900
LENSYM = LENCST(MINSYM,4)
CALL NOVCST(MINSYM,4),(-4),
  - MINSYM.(LENSYM).(-LENSYM)." ' "  B RIGHT JUSTIFY
LENSYM = LENCST(MAXSYM,4)
CALL NOVCST(MAXSYM,4),(-4),
  - MAXSYM.(LENSYM).(-LENSYM)." ' "  B RIGHT JUSTIFY
WRITE(6,385) MINSYM,MINNUM,MAXSYM,MAXNUM
FORMAT(' SYMBOLS. ',A4,' ',J3,' ',A4,' ',J3)
GO TO 900

C
C NO SYMBOLS SPECIFIED -- CONFIRM ALL SYMBOLS
C
850 IF(MCFIRM.NE.0) CALL ALLSYM
C
C NORMAL RETURN
C
900 KOMO=' '
RETURN

C
C
C INTERNAL
C
C SUBROUTINE ALLSYM 8 CONFIRM ALL CURRENT SYMBOLS
C
INTEGER NXTSYM 8 NEXT SYMBOL
C
MINNUM = 0
CALL NOVCST(MINSYM,1),(6), KSYM(1),(1),(4)," '
DO 200 MAXNUM = 0, ISYM
  CALL NOVCST(NXTSYM,1),(6), KSYM(MAXNUM+2),(1),(4)," '
  IF(NXTSYM.EQ.MINSYM) GO TO 200 8 NEXT SYM SAME AS CURRENT
  IF(MINSYM.EQ."$$$") MINSYM='?' 8 UNDEFINED SYMBOL
LENSYM = LENCST(MINSYM,4)
CALL NOVCST(MINSYM,4),(-4),
  - MINSYM.(LENSYM).(-LENSYM)." ' "  B RIGHT JUSTIFY
& WRITE(6,145) MINSYM,MINNUM
145 FORMAT(' SYMBOLS. ',A4,' ',J3)
& IF(MINNUM.NE.MAXNUM)
& WRITE(6,165) MINSYM,MINNUM,MAXNUM
165 FORMAT(' SYMBOLS. ',A4,' ',J3,' ',A4,' ',J3)
MINNUM = MAXNUM + 1
MINSYM = NXTSYM
200 CONTINUE
RETURN
OAN PACKAGE APPENDIX M
COMMAND ROUTINES

END
SUBROUTINE KNXTAB 8 TABULATE FREQUENCY OF DATA IN PREVIOUS WINDOW
U NOH) 8 1: FIRST 3 CHARS OF COMMAND 0: SPACES

HISTORY

E H SCHLOSSER LEC 05/20/74 ORIGINAL CODE
E H SCHLOSSER LEMSCO 05/20/80 CHG NAME FROM DISTAB & ADD COL/INT

METHOD

TABULATE BY RADIANCE, DENSITY, CLASS, SYMBOL, COLOR, OR INTENSITY.

MACHINE-DEPENDENT CODE

ASSUMES AT LEAST 6 CHARACTERS PER KSYM ELEMENT.

EXTERNAL REFERENCES

MONOTE 8 PRINT/COUNT/LOG 'NOTE' DIAGNOSTIC MESSAGE
MOCLRM 8 CLEAR 'WARNING' DIAGNOSTIC
IDENTify 8 IDENTIFY ERTS SCENE
MOVES 8 MOVE CHARACTER STRING
GETICE 8 GET INTEGER-CHARACTER-EQUIVALENT
CHARINDEX 8 CHARACTER STRING FOR INTEGER
COLORI 8 COLOR NAME (STRING) FOR INTEGER-COLOR-EQUIVALENT

EXCEPTIONS

1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
COLOR OR INT SPEC. BUT COLOR MODE SWITCH (MCOLOR IN KOMXQT) NOT ON
2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KOMXQT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KORNER.LIST 8 COMMON ERTS PARAMETERS
INCLUDE KOKLKS.LIST 8 COMMON CLASSIFICATION INFO
INCLUDE KOMSYM.LIST 8 COMMON SYMBOL TABLE
INCLUDE KORTBL.LIST 8 COMMON MULTI-PURPOSE TABLE
INCLUDE NULCST.LIST 8 DEFINE NULL CHARACTER STRING
INCLUDE MAXINT.LIST 8 DEFINE MAXIMUM INTEGER

M-182
C LOCAL DECLARATIONS

C

INTEGER INTEHP TEMPORARY
INTEGER KTSTYP TABULATION TYPE (3 CHARs)
INTEGER NSYTYP NUMBER OF SYMBOL TYPE:
  1 = CHARACTER STRING SYMBOL
  2 = INTENSITY SYMBOL
  3 = COLOR SYMBOL
INTEGER NLCSYT 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/DEM FOR CURRENT SYMBOL
INTEGER KFRSSTR(8).KFRTOS(8) 8 FREQ SUB-TOTALS. TOTALS

C

C PROCEDURE

C

CALL TRACE

C CHECK IF KOMTBL LOADED WITH FREQUENCY INFO

IF(KLSTYP.EQ.0).OR. 8 NO CLASS TYPE FROM PREVIOUS DISPLAY
  & (KTBLTY.NE.'FREQ') 8 NO FREQUENCY TABLES LOADED FROM PREVIOUS DISPLAY
  & CALL HOMARN( 'NO PREVIOUS WINDOW TO TABULATE')

C GET/CHECK TABULATE TYPE

KTBTYP='SYM' 8 DEFAULT IS TABULATION OF CHARACTER STRING SYMBOLS
NSYTYP=MAXINT 8 NOT YET DEFINED
CALL GETSKM(KTBTYP,3), HULCSIT
IF(KTBTYP.EQ.KLSTYP) NSYTYP=1
IF(KTBTYP.EQ.'SYM') NSYTYP=1
IF(MCOLOR.NE.0).AND.(KTBTYP.EQ.'INT') NSYTYP=2
IF(MCOLOR.NE.0).AND.(KTBTYP.EQ.'COL') NSYTYP=3
NLCSYT=1
IF(NSYTYP.EQ.3) NLCSYT=2
IF(NSYTYP.EQ.MAXINT) CALL WARN( 'BAD TABULATE TYPE ---')

C DRAIN SPECS FOR CURRENT COMMAND

CALL GETSIN(1TEMP. 'EXTRA TABULATE SPECIFICATION ---')

C CHECK FOR DIAGNOSTICS

IF(MOTAC.NE.0) GO TO 800 8 DATA/CHECKOUT MODE
IF(MOTORL.NE.0) GO TO 800
C USE PREVIOUS WINDOW NUMBER FOR PAGE HEADING
C
WNDTEMP=WINDNUM
8 SAVE CURRENT WINDOW NUMBER
WNDNUM=KTBNEW
8 WINDOW NUMBER FROM FREQUENCY TABLES
C
C PRINT PAGE/WINDOW HEADINGS
C
WRITE(9,105) WINDNUM,MTERAL
105 FORMAT(' WINDON NUMBER ',J3,'.'G8.4,'TABULATE','.'G8.4)
CALL MOUNT(9,10)
WRITE(16,105) WINDNUM,MTERAL

C RESTORE CURRENT WINDOW NUMBER
C
WNDNUM=WNDTEMP
C
C PRINT TABLE HEADINGS
C
CALL IDENT(6)
CALL IDENT(10)
IF(INHATCH.EQ.0)
6 WRITE(6,128) LINCH(NLCSY1),LINCH(N),NLINC
WRITE(10,128) LINCH(NLCSY1),LINCH(N),NLINC
128 FORMAT(6 'SYMBOL RADIANCE FREQUENCY' /
6 'CH',12,' VALUE ',5(F15.1))
WRITE(6,135)
WRITE(10,135)
135 FORMAT(1X)

C
C INITIALIZE SUB-TOTALS, TOTALS & LIMITS
C
DO 340 N=1,5
KFRSBS(N)=0
KFRTOT(N)=0
340 CONTINUE
MINCHV=MAXINT
MAXCHV=MAXINT
DO 360 N=1,NLINC
MINCHV=MIND(MINCHV,LIVD(N))
MAXCHV=MAXD(MAXCHV,LIVD(N))
360 CONTINUE
MINRAD=MINCHV
CALL MOVCST(MINSYM1,11,3)
CALL MOVCST(MINSYM1,10,8)
CALL MOVCST(MINSYM1,10,3)
CALL GETICE(INTEMP, KSYM,MINRAD=11,9)
CALL CSTY4(MINSYM2,12,9, INTEMP=110.4)
CALL PUTCHR(MINSYM2,51, 51)
CALL GETICE(INTEMP, KSYM,MINRAD=11,6)
CALL KOLR4(MINSYM3, INTEMP)

C
PRINT TABULATIONS

DO 600 MAXRAD=MINTCHV,MAXCHV
DO 410 N=1,NLIMCH
  KFRSUB(N)=KFRSUB(N)+KFREQ(MAXRAD+1,N)
410 CONTINUE
  CALL MOVCT(NEXSYM(1),(1,13),",",(1,1,1,1)
  CALL MOVCT(NEXSYM(1),(41,8)," KSYM(MAXRAD+1,(1,1,1)
  CALL MOVCT(NEXSYM(1),(101,13)," KSYM(MAXRAD+2),(2,1,3)
  CALL GETICE(INTEMP, KSYM(MAXRAD+1,8))
  CALL CSTIN(NEXSYM(1),(11,(12),(INTEMP+1)<10,4)
  CALL PUTCHR(NEXSYM(1),(5),"")
  CALL GETICE(INTEMP, KSYM(MAXRAD+1,8))
  CALL KOLR(NEXSYM(1),(NLIMCH)
IF(KSOLV(NE,KSOLV(1,1)," SG AGGREGATE TABULATIONS BY SYMBOL"
  IF(MAXRAD.LT.LCYNV11) AND."
  CALL MOVCT(NEXSYM(1),(1,1,1,1))
  CONTINUE
  IF(MAXRAD.EQ.MAXRAD) GO TO 440
  IF(NBATCH.EQ.0)
    WRITE(6,428) MINTSYM(NTYP),MINRAD,MAXRAD,
          (KFRSUB(N),N=1,NLIMCH)
    WRITE(10,429) MINTSYM(NTYP),MINRAD,MAXRAD,
          (KFRSUB(N),N=1,NLIMCH)
  FORMAT(1X,A9,1X,J3,1X,J3,916)
  GO TO 480
440 IF(NBATCH.EQ.0)
  WRITE(6,448) MINTSYM(NTYP),MINRAD,(KFRSUB(N),N=1,NLIMCH)
  WRITE(10,449) MINTSYM(NTYP),MINRAD,(KFRSUB(N),N=1,NLIMCH)
  FORMAT(1X,A9,1X,J3,916)
480 IF(NCSYM.LT.2) GO TO 590
  DO 990 NCHAR=2,NCSYM 8 OVERPRINT SYMBOLS
    CALL GETCHR(KNTEMP, MINTSYM(NCHAR-1))
    IF(KNTEMP.NE."") WRITE(10,479) KNTEMP
    FORMAT(1X,91,3X,A1)
590 CONTINUE
590 DO 970 N=1,5
  KFRTOT(N)=KFRTOT(N)+KFRSUB(N)
  KFRSUB(N)=0
970 CONTINUE
  MINRAD=MAXRAD+1
  CALL MOVCT(MINTSYM(11),(138), MINTSYM(11),138)
  CONTINUE
C
C PRINT TOTALS
C
  IF(NBATCH.EQ.0)
IF(NBATCH.EQ.0)
    WRITE(6,818) (KFRTOT(N),N=1,NLIMCH)
    WRITE(10,818) (KFRTOT(N),N=1,NLIMCH)
  FORMAT(1X,12X,17,918)
  GO TO 800
818 FORMAT(1X,12X,17,416)
C
C 800 CONTINUE
DAN PACKAGE APPENDIX M
COMMAND ROUTINES

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(6,615)
615 FORMAT('...TRY AGAIN')
   GO TO 890
820 IF (INDTAL.EQ.0) GO TO 850
   CALL MDNOTE('PREVIOUS FATAL ERRORS -- NO TABULATION GENERATED')
   GO TO 890
850 IF (MCHECK.EQ.0) GO TO 890
   CALL MDNOTE('CHECKOUT MODE -- NO TABULATION GENERATED')

C CLEAR WARNINGS
C
890 CALL MDCLRWIN(NULCST)

C RETURN
C
900 KOMD=" 
   RETURN
END
SUBROUTINE KMOTIC - GET/CHECK WINDOW TICK INTERVALS
U KONG - FIRST 3 CHAR OF COMMAND O: SPACES

HISTORY
-------
E H SCHLOSSER  LEC  09/12/73  NUMERIC OPTION FOR WINDOW CENTER
E H SCHLOSSER  LEC  09/06/75  ALPHANUMERIC COMMAND FOR ORIGIN
E H SCHLOSSER  LEC  07/01/78  DELETE RETURN K
E H SCHLOSSER  LEC  02/29/79  REVISE GETS CALLS
E H SCHLOSSER  LEC  12/12/79  UPGRADE DOCUMENTATION

METHOD
-------
GET TICK COORDINATE SYSTEMS & INTERVALS, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE
------------------------
ASSUMES 8 CHARACTERS PER INTEGER.
ASSUMES WORD STARTING WITH ALPHA CHARACTER, WHEN INTERPRETED AS AN INTEGER, IS POSITIVE!

EXTERNAL REFERENCES
---------------------
SPANS 8 ENABLE SPECIFICATIONS TO SPAN CARDS
GET5KX 8 GET CHARACTER STRING DATA FIELD FROM UNIT 5
WARN5 8 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
GET5IN 8 GET INTEGER DATA FIELD FROM UNIT 5
GET5SX 8 GET SEXAGENARY DATA FIELD FROM UNIT 5
GET5RL 8 GET REAL DATA FIELD FROM UNIT 5

EXCEPTIONS
--------
1. THE FOLLOWING CONDITIONS GENERATE 'WARNING' DIAGNOSTICS:
   COORDINATE SYSTEM NOT SCANT (DEGREES) OR MINUTES (K)
   OR METRES!

2. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   2 <= LINE <= 3000
   2 <= SAMPLE <= 4000
   .001 <= LATITUDE <= 3.
   .001 <= LONGITUDE <= 3.
   1E+2 <= EASTING <= 5E+5
   1E+2 <= NORTHING <= 5E+5

M-127
3. AN EXTRA SPECIFICATION GENERATES A "MANNING" DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KOMXOT.LIST
INCLUDE KOMIWW.LIST
INCLUDE KOMOWW.LIST
INCLUDE WINDEF.LIST
INCLUDE NULCST.LIST

LOCAL DECLARATIONS

REAL RLTEMP(2)   # TEMPORARY
INTEGER JTYPE(2)  # TYPE OF TICKS: 'PRIMARY' OR 'SECONDARY'

PROCEDURE

CALL TRACE

CALL SPANS( 2)   # ALLOW INPUT FOR THIS COMMAND TO SPAN 2 CARDS
KMD='TICK.
JTYPE(1)='PRIMA'
JTYPE(2)='RY.'
MOD=WTIC   # START WITH PRIMARY TICKS

GET COORDINATE SYSTEM

200 KORDSY=KSYOWW(NOD)   # USE OLD COORDINATE SYSTEM IF NONE SPECIFIED
CALL GETSKH(KORDSY,(3), NULCST)
IF(KORDSY.GT.0) GO TO 220   # POSITIVE WORD STARTS WITH ALPHA CHARACTER

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

CHECK COORDINATE SYSTEM

220 IF(KORDSY.EQ."CA") GO TO 300
IF(KORDSY.EQ."DE") GO TO 400
IF(KORDSY.EQ."MIN") GO TO 410
IF(KORDSY.EQ."KM") GO TO 600
IF(KORDSY.EQ."ME") GO TO 610

M-128
200 CALL WARN5( 'BAD COORDINATE SYSTEM --' )
       GO TO 800

C SCANER (LINE-LENGTH ADJUSTED) COORDINATES

300 CALL GETSIN(MSAOWW(WLIN.NOD)), 2,3000.'BAD LINES --')
       CALL GETSIN(MSAOWW(WSAM.NOD)), 2,4000.'BAD SAMPLES --')
340 IF(MCFIRM.NE.0) WRITE(6,345)
       & KMO.MSAOWW(WLIN.NOD).MSAOWW(WSAM.NOD),J TYPE
649 FORMAT(I9.A6,' SCAN. ',14,' LINES. ',14,' SAMPLES'.2A6)
       GO TO 800

C GEOGRAPHIC COORDINATES

400 CF=1.0 DEGREES -- NO CONVERSION NEEDED
       GO TO 420
410 CF=1.0/60. DEGREES -- NO CONVERSION NEEDED
420 CALL GETSIX(ODEOWW(WLAT.NOD)), CF, 001.3.'BAD LATITUDE --'
       CALL GETSIX(ODEOWW(WLON.NOD)), CF, 001.3.'BAD LONGITUDE --'
       IF(KORDSY.EQ.'MIN') GO TO 450
       IF(MCFIRM.EQ.0) GO TO 800
       WRITE(6,445) KMO.ODEOWW(WLAT.NOD).ODEOWW(WLON.NOD),J TYPE
445 FORMAT(I9.A6.' DEGREES. ',F7.5,' LAT. ',F7.5,' LON'.2A6)
       GO TO 800
450 RLTMP(1)=ODEOWW(WLAT.NOD)*60.
       RLTMP(2)=ODEOWW(WLON.NOD)*60.
       IF(MCFIRM.EQ.0) WRITE(6,455) KMO.RLTMP,I TYPE
455 FORMAT(I9.A6.' MINUTES. ',F7.3,' LAT. ',F7.3,' LON'.2A6)
       GO TO 800

C UTM COORDINATES

600 CF=1E+3 A CONVERT FROM KILOMETRES TO METRES
       GO TO 620
610 CF=1.0 METRES -- NO CONVERSION NEEDED
620 CALL GETSRL(UTMOWW(WEA.NOD)), CF,1E+2.5E+5.'BAD EASTING --'
       CALL GETSRL(UTMOWW(WNO.NOD)), CF,1E+2.5E+5.'BAD NORTHING --'
       IF(MCFIRM.EQ.0) WRITE(6,645)
       & KMO.UTMOWW(WEA.NOD).UTMOWW(WNO.NOD),J TYPE
645 FORMAT(I9.A6.' KM. ',-3P.F6.3,' EAST. ',F6.3,' NORTH'.2A6)

C STORE COORDINATE SYSTEM

800 KSYOWW(NOD) = KORDSY
       IF(MOD.NE.MTIC) GO TO 890
       NOD=MTIC+1 & READY FOR SECONDARY TICKS
       KMO=" & DON'T PRINT COMMAND WITH SECONDARY TICKS
       JTYPE(1)=' SECON'
       JTYPE(2)='DARY'
       GO TO 200
890 CALL GETSIN(ITEMP, *1,-1,'EXTRA TICK SPECIFICATION --')
DAM PACKAGE APPENDIX M
COMMAND ROUTINES

C
C NORMAL RETURN
C
900 KOND=" 
RETURN
END
SUBROUTINE KMOTIM: PRINT CLOCK TIME & CHARGE TIME
U: KOMD) & I: FIRST 3 CHAR OF COMMAND & O: SPACES

HISTORY

E H SCHLOSSER LEW 03/02/79 ROMTS/DESIGN/ CODE
E H SCHLOSSER LEW 11/26/79 UPGRADE DOCUMENTATION

METHOD

GET TIMES FROM OPERATING SYSTEM & PRINT THEM.

MACHINE-DEPENDENT CODE

UNIVAC EXEC-8 EXECUTIVE REQUESTS.

EXTERNAL REFERENCES

GETSIN: GET INTEGER DATA FIELD FROM UNIT 5
ERDATE: RETURN SYSTEM DATE & TIME
ERSUPS: RETURN ACCUMULATED SUP TIME IN 200 USEC INCREMENTS
MVCST: MOVE CHARACTER STRING
RL215X: INTEGER SEXTENARY ARRAY FROM REAL

EXCEPTIONS

1. COMMAND SPECIFICATIONS ARE NOT ALLOWED.
2. KOMD IS NOT CHECKED FOR VALIDITY.

GLOBAL DECLARATIONS

NONE.

LOCAL DECLARATIONS

INTEGER INOY: MONTH-DAY-YEAR (6 CHARACTERS)
INTEGER IMHS: HOUR-MINUTE-SECOND (8 CHARACTERS)
INTEGER IM: MINUTE (2 CHARACTERS)
INTEGER IS: SECOND (2 CHARACTERS)
INTEGER MS200: CHARGE TIME IN 200 MILLISECOND INCREMENTS
INTEGER TIME(3) 8 HOURS/MINUTES/SECONDS OF CHARGE TIME
REAL FSECS 8 FRACTIONAL SECONDS OF CHARGE TIME

C
C
C PROCEDURE
C --------
C
C
C CALL TRACE
C
CALL GETSIN(NMRS, +1,-1,'NO TIME SPECIFICATION ALLOWED --')
CALL ERDATE(INDY,IMNS )
CALL MOVCS(TIM,(1),(2), IMMS,(3),(2),', ')
CALL MOVCS(TIM,(1),(2), IMMS,(3),(2),', ')
CALL ERSUPS(MS200 )
SECS=FLOAT(MS200)/1000000.
CALL RL2ISX(SECS,TIME,3,FSECS)
WRITE(6,145)
& IMNS,IM,TS.
& TIME,FSECS
145 FORMAT(TIME, '/
 & (CLOCK) ',A2,' : ',A2,' : ',A2,'/
 & (CHARGE) ',J2,' : ',J2,' : ',J2,F4,3)
C
KOMO="" RETURN
END
NOT IMPLEMENTED
SUBROUTINE KMOTOLI & GET/CHECK TOLERANCE

**U KOMD1** 8 1: FIRST 3 CHAR OF COMMAND 0: SPACES

---

**HISTORY**

E M SCHLOSSER LEC 08/10/78 ORIGINAL CODE
H A TOMPKINS LEMSCO 09/27/80 UPGRADE DOCUMENTATION

**METHOD**

UPDATE TOLERANCE FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

**MACHINE-DEPENDENT CODE**

NONE.

**EXTERNAL REFERENCES**

GETSIN & GET INTEGER DATA FIELD FROM UNIT 5

**EXCEPTIONS**

1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE 'WARNING' DIAGNOSTICS:
   
   | TOLERANCE | = 6 |

2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

**GLOBAL DECLARATIONS**

INCLUDE KOMQT.LIST & COMMON PROGRAM EXECUTION SWITCHES, COUNTERS

INCLUDE KOMCLS.LIST & COMMON CLASSIFICATION INFO

**LOCAL DECLARATIONS**

NONE.

**PROCEDURE**

M-134
CALL TRACE

LCVTOL=MAX(0,LCVTOL)
LCVTOL=MIN(0,LCVTOL)
CALL GETSIN(LCVTOL, 0,0,'BAD TOLERANCE -->')
CALL GETSIN(IDUMMY, 01,1,'EXTRA TOLERANCE SPECIFICATION -->')
IF(INCFIRM.NE.0) WRITE(6,129) LCVTOL
129 FORMAT(' TOLERANCE: ',11)
KMOD=''
RETURN
END
SUBROUTINE KMONIN: GET/CHECK WINDOW VERTICES

U KOND) 6 1: FIRST 3 CHARS OF COMMAND 01 SPACES

HISTORY

E H SCHLOSSER LEC 08/09/73 NUMERIC OPTION
E H SCHLOSSER LEC 12/07/79 ALPHANUMERIC COMMAND
N A TOMPKINS LEMSCO 09/27/80 UPGRADE DOCUMENTATION

METHOD

UPDATE WINDOW VERTICES FROM UNIT 5, IF SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE

USES UNIVAC FORTRAN V RETURN K.

EXTERNAL REFERENCES

MDWARN 6 PRINT/COUNT/LOG 'WARNING' DIAGNOSTIC MESSAGE
UNGETS 6 BACK UP 1 FIELD ON UNIT 5
GETSKH 6 GET CHARACTER STRING DATA FIELD FROM UNIT 5
GETSN 6 GET INTEGER DATA FIELD FROM UNIT 5
GETSR 6 GET REAL DATA FIELD FROM UNIT 5
GETSIX 6 GET SEXAENARY DATA FIELD FROM UNIT 5
WARNS 6 GENERATE WARNING FOR INVALID/MISSING FIELD FROM UNIT 5
GETSN 6 GET INTEGER DATA FIELD FROM UNIT 5
SPANS 6 ENABLE SPECIFICATIONS TO SPAN CARDS
WRVERT 6 WRITE VERTEX COORDINATES FOR WINDOW

EXCEPTIONS

1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE
   'WARNING' DIAGNOSTICS:
   -2500 <= SCANNER LINE <= 2500
   -3500 <= SCANNER SAMPLE <= 3500
   -3. <= LATITUDE DEG <= 90.
   -3. <= LONGITUDE DEG <= 180.
   -1E-6 <= EASTING METRES <= 1E-6
   -1E-6 <= NORTING METRES <= 9E-6
   -800 <= PRINT LINES <= 800
   -800 <= PRINT COLUMNS <= 800

2. TOO MANY VERTICES GENERATE A 'WARNING' DIAGNOSTIC.
GLOBAL DECLARATIONS

```
INCLUDE KOMXOT.LIST  # COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMINW.LIST  # COMMON INPUT WINDOW Packets
INCLUDE KOMOUT.LIST  # COMMON OUTPUT WINDOW Packets
INCLUDE WINDF.LIST   # DEFINE STRUCTURE OF WINDOW Packets
INCLUDE NULCST.LIST  # DEFINE NULL CHARACTER STRING
```

LOCAL DECLARATIONS

```
PARAMETER NODEMIN = KVVER+1 # NODE MINIMUM
INTEGER KORSY    # COORDINATE SYSTEM
INTEGER JCOMMA   # COMMA/BLANK DEPENDING ON POSITION IN PRINT LINE
INTEGER NODE    # NODE
INTEGER N       # NODE COUNTER
INTEGER KONAME  # COORDINATE NAME
REAL CFL       # CONVERT FROM XXX TO PRINT LINES
REAL CRC       # CONVERT FROM XXX TO PRINT COLUMNS
REAL CR       # CONVERSION FACTOR
INTEGER MP      # MAXIMUM # OF PLACES
```

PROCEDURE

```
CALL TRACE
```

INITIALIZE

```
CALL SPANS(101) # ALLOW INPUT FOR THIS COMMAND TO SPAN UP TO 10 CARDS
NOD=KVVER  # SET NODE POINTER BEFORE FIRST VERTEX
```

CHECK COORDINATE SYSTEM

```
KORSY=KSYOWH(KVVER)  # USE OLD COORDINATE SYSTEM IF NONE SPECIFIED
CALL GETSKM(KORSY,'SC'), NULCST:
IF(KORSY.EQ.'SC') 00 TO 300  # SCANNER COORDINATES
IF(KORSY.EQ.'DEG') 00 TO 400  # GEOGRAPHIC COORDINATES IN DEGREES
IF(KORSY.EQ.'MIN') 00 TO 410  # GEOGRAPHIC COORDINATES IN MINUTES
IF(KORSY.EQ.'KM') 00 TO 600  # UTM COORDINATES IN KILOMETRES
IF(KORSY.EQ.'MET') 00 TO 810  # UTM COORDINATES IN METRES
IF(KORSY.EQ.'PR') 00 TO 700  # MAP COORDINATES IN PRINT LINES & COLUMNS
IF(KORSY.EQ.'CM') 00 TO 705  # MAP COORDINATES IN CENTIMETRES
IF(KORSY.EQ.'IN') 00 TO 710  # MAP COORDINATES IN INCHES
CALL WARNS("BAD COORDINATE SYSTEM --")
00 TO 999
```

SCANNER (LINE-LENGTH ADJUSTED) COORDINATES

M-137
COMMAND ROUTINES

C
300 CALL CXCLOS(INSAWMN.3340.3900)
      CALL GTSI(INSAWMN.IN.NODI. -3500.8900. "BAD LINE --")
      CALL GTSI(INSAWMN.ISAN.NODI. -3500.3900. "BAD SAMPLE ----")
      GO TO 300
340 WRITE(8.345)!
341 P storage WINDOW. SCAN.
      JCOMMA=.
360 DC 360 N=NSDINN.NODD
      IF (N.2.00) JCOMMA=.
      WRITE(8.386) (INSAWMN.I.1=1.8), JCOMMA
380 CONTINUE
      GO TO 900

C

C GEOGRAPHIC COORDINATES

400 KONAME='DEGREE'
      CF=1. 0 DEGREES -- NO CONVERSION NEEDED
      MP=3  0 MAX NO OF SEX'Y PLACES
      GO TO 420
410 KONAME='MINUTE'
      CF=1./60. 0 CONVERT FROM MINUTES TO DEGREES
      MP=2  0 MAX NO OF SEX'Y PLACES
      CALL CXXS(X(INSAWMN.6440.5900)
      CALL GTS(X(INSAWMN(INLAT.NODI. +1E+4.1E-6. "BAD LATITUDE --")
      CALL GTS(X(INSAWMN(INLAT.NODI. +1E+4.1E-0. "BAD LONGITUDE --")
      GO TO 420
440 WRITE(8.449) KONAME
      CALL WAVERT(INSAWMN.CF.CF.'13X.FB.FB.4.6H LAT. FB.4.4H LON.A11'.NODI)
      GO TO 900

C

C UTM COORDINATES

600 CF=1E+3  0 CONVERT FROM KILOMETRES TO METRES
      GO TO 620
610 CF=1. 0 METRES -- NO CONVERSION NEEDED
620 CALL CXXS(UXMOMA.6440.6900)
      CALL GTS(UXMOMA(ME4.NODI. +1E+4.1E-6. "BAD EASTING --")
      CALL GTS(UXMOMA(ME4.NODI. +1E+0.1E-6. "BAD NORTHING --")
      GO TO 620
640 WRITE(6.649) KORDXY
649 FORMAT(13X.19. LINE. "19. SAMPLE")
      CALL WAVERT(UXMOMA.CF.CF.
      8 '13X.FB.FB.3.7H EAST. FB.3.6H NORTH.A11'.NODI)
      GO TO 900

C

C PRINT/ PLOT COORDINATES

700 KONAME='PRINT'
      CFL=1. 0 PRINT LINES -- NO CONVERSION NEEDED
      CFC=1. 0 PRINT COLUMNS -- NO CONVERSION NEEDED

R-139
GO TO 720
700 KONAME='KORDSY'
  CFL=0.3937*LINCH 8 CONVERT FROM CENTIMETRES TO PRINT LINES
  CFL=0.3937*KINCH 8 CONVERT FROM CENTIMETRES TO PRINT COLUMNS
GO TO 720
710 KONAME='INCHES'
  CFL=FLOAT(LINCH) 8 CONVERT FROM INCHES TO PRINT LINES
  CFL=FLOAT(KINCH) 8 CONVERT FROM INCHES TO PRINT COLUMNS
720 CALL CKCLOSE(PPDWNM,PPDOWN,IN,NOD).
  CFL=888..888..'**BAD LINE --''
  CALL GETSR(PPDWNM,NOD).
  CFL=888..888..'**BAD COLUMN --''
GO TO 720
740 WRITE(6,740) KONAME
740 FORMAT(' WINDOW, '..AB..')
  IF(KORDSY.NE..'PRI') GO TO 760
  CALL HRVERT(PPDWNM,CFL,CFC).
  6 '13%,F8.7,7H LINE,..FS,0.T8,7H COLUMN,A11',NOD)
GO TO 900
780 CALL HRVERT(PPDWNM,CFL,CFC).
  6 '13%,F8.7,7H LINE,..FS,0.T8,7H COLUMN,A11',NOD)

C NORMAL RETURN
C C 900 KONAME=''
  RETURN
C
C
SUBROUTINE CKCLOSE 8 CLOSE POLYGON IF NEXT VERTEX IS ABSENT
8 INTOWN, 8 WINDOW PACKET
8 S, 8 CLOSE WITH CONFIRMATION TRANSFER LABEL
8 S 8 CLOSE WITHOUT CONFIRMATION TRANSFER LABEL
C
DIMENSION INTOWN(2,1)
C
KNTEMP=' NUL'
  CALL .GETSK(KNTEMP(1),KNTEMP(1),NULCST)
  IF(KNTEMP.EQ..' NUL') GO TO 110 8 MORE SPECS
  CALL UNGETS
  NOD=NOD+1 8 READY TO GET NEXT VERTEX
  IF(NOD.LE.WH0) RETURN
  CALL WAMARN(' TOO MANY VERTECES -- EXCESS IGNORED')
  NOD=NOD-1
110 IF(NOD.NE.WVER) GO TO 150
NOD=INTOWN(MUSED,WHEAD) 8 NO VERTICES SPECIFIED -- USE OLD WINDOW
GO TO 900
150 INTOWN(MUSED,WHEAD)=NOD 8 UPDATE HIGHEST NODE USED
  INTOWN(1,WVER)=INTOWN(1,NOD) 8 CLOSE ORDI\NATE
  INTOWN(2,WVER)=INTOWN(2,NOD) 8 CLOSE ABSOSSA
  NKLAST=INTOWN-11+2 8 UPDATE POINTER TO ABSOSSA OF NEXT-TO-LAST VERTEX
C
C
H-139
OAR PACKAGE APPENDIX M
COMMAND ROUTINES

KSYOWN(MVER)=KdROS
IF(INCFIRN.EQ.0) RETURN 3
   0 CLOSED -- DON'T CONFIRM VERTICES
RETURN 2
   0 CLOSED -- CONFIRM VERTICES
END

M-140
SUBROUTINE KMDXXX U KOM01 8 1: FIRST 3 CHARS OF PROG NAME PLUS ‘ ‘ 0: SPACES IF VALID

HISTORY

E H SCHLOSSER  LEC  01/17/79  ORIGINAL CODE

METHOD

GET COMMAND. IF IT IS 8 CHARACTERS OR SHORTER, PREFIX KOM0 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 KMXXEO AND CHANGE KOMO TO SPACES.

MACHINE-DEPENDENT CODE

UNIVAC EXEC 8 PROGRAM FILE NAMING CONVENTIONS.

EXTERNAL REFERENCES

UNGETS  BACK UP 1 FIELD ON UNIT 5
PUTCHR  PUT CHARACTER INTO CHARACTER STRING
GETSKH  GET CHARACTER STRING FIELD FROM UNIT 5
LENCST  GET LENGTH OF CHARACTER STRING
MOVCS  MOVE CHARACTER STRING
LOCDSF  LOCATE DISK SYMBOLIC FILE OR ELEMENT
KMXXED  EDIT ACTUAL SPECS INTO MACRO COMMAND DEFINITION

EXCEPTIONS

1. KOMO 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.

GLOBAL DECLARATIONS

INCLUDE KOMKST.LIST  3 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE NULCST.LIST  DEFINE NULL CHARACTER STRING

M-141
LOCAL DECLARATIONS

INTEGER MAKOMO(3)
INTEGER NAMELT(2)
INTEGER NAMFIL(2)
INTEGER LOCFIL
INTEGER LENGTH

PROCEDURE

CALL TRACE

GET 1ST 18 CHARACTERS OF MACRO COMMAND

CALL UNOETS
CALL PUTCHR(MAKOMO.(18), '7')  & INITIALIZE MAKOMO TO 18 CHAR LENGTH
CALL GETSKH(MAKOMO.(18), NULCST)

CHECK LENGTH OF MACRO COMMAND

LENGTH=LENCST(MAKOMO.18)
IF(LENGTH.LT.3) GO TO 900  & TOO SHORT
IF(LENGTH.GT.12) GO TO 900  & TOO LONG

CONSTRUCT NAME OF ELEMENT CONTAINING TEXT FOR MACRO COMMAND

NAMELT=KOMO  & 1ST 3 CHARS OF PROGRAM NAME PLUS '
IF(LENGTH.LE.8) CALL MOVCST(NAMELT.111,8), MAKOMO.(1),(8),<KOMD><KOMD>
         MAKOMO(111,12).
IF(LENGTH.GT.8) CALL MOVCST(NAMELT.111,12), MAKOMO.(111,12),<KOMD><KOMD>

CHECK IF ELEMENT WITH TEXT FOR MACRO-COMMANO IS ON DISK

NAMFIL='MACOAM'
LOCFIL=LOCDSF(NAMFIL.NAMELT,')
IF(LOCFIL.LE.0) NAMFIL='OAM'
IF(LOCFIL.LE.0) LOCDSF(NAMFIL.NAMELT,')
IF(LOCFIL.LE.0) GO TO 900  & NO SYMBOLIC ELEMENT FOR MACRO COMMAND

EDIT ACTUAL SPECS INTO MACRO COMMAND DEFINITION

CALL KMXXED(MAKOMO,NAMFIL.NAMELT,LOCFIL)
DONE

900 RETURN
END
SUBROUTINE KNDZON: GET/CHECK UTM PROJECTION ZONE
U KONDO: 6 CHAR OF COMMAND 0: SPACES

HISTORY

- E M SCHLOSSER LEC 10/29/75 ORIGINAL CODE
- E M SCHLOSSER LEC 07/30/78 DELETE RETURN K
- E M SCHLOSSER LEC 02/20/79 REVISE GETS, CALLS
- E M SCHLOSSER LEC 11/29/79 UPGRADE DOCUMENTATION

METHOD

UPDATE UTM CENTRAL MERIDIAN FROM UNIT 5, IF ZONE SPECIFIED, AND CONFIRM.

MACHINE-DEPENDENT CODE

NONE.

EXTERNAL REFERENCES

GETSIN: GET INTEGER DATA FIELD FROM UNIT 5

EXCEPTIONS

1. SPECIFICATIONS OUTSIDE THE FOLLOWING RANGES ARE REJECTED AND GENERATE 'WARNING' DIAGNOSTICS:
   \[ 1 \leq \text{ZONE} \leq 22 \]

2. AN EXTRA SPECIFICATION GENERATES A 'WARNING' DIAGNOSTIC.

GLOBAL DECLARATIONS

INCLUDE KOMXEQ, LIST COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMFIT, LIST COMMON ADJUSTMENT/REGISTRATION PARAMETERS
INCLUDE KOMOWW, LIST COMMON OUTPUT WINDOW PACKETS
INCLUDE WINDEF, LIST DEFINE STRUCTURE OF WINDOW PACKETS

LOCAL DECLARATIONS

REAL OLOCMO A VALUE OF UTMCMO ON ENTRY (BEFORE POSSIBLE CHANGING)
INTEGER INTEMP A TEMPERARY
C

C PROCEDURE
C

C CALL TRACE
C

IF(UTMCMOD.EQ.0.1 UTMCMOD=-35817. I \ INVALID (ZONE = 6000)
OLDCMD=UTMCMOD
INTEMP=(183.-UTMCMOD)/6. +.5
CALL GETSIN(INTEMP, 1.22, 'BAD UTM ZONE --')
UTMCMOD=183.-6*INTEMP
IF(MCFIRM.NE.0) WRITE(6,125) INTEMP,UTMCMOD
125 FORMAT(' ZONE, ',12,' (UTH CENTRAL MERIDIAN ',F9.1, ' DEG)')
CALL GETSIN(INTEMP, +1,-1, 'EXTRA ZONE SPECIFICATION --')
IF(OLDCMD.NE.UTMCMOD) KSYOHW(NORMO)= 'NUL' & IF CMD CHANGED MARK ORIG ABSENT
900 KOMO=' ' RETURN
END

SUBROUTINE KMOBAD( O SADD -- DYNAMIC SADD
 U KOMO) & 1: FIRST 3 CHARS OF COMMAND O: SPACES

C (E H SCHLOSSER)

C
C INCLUDE KOMOGT.LIST
INCLUDE ASMDEF.LIST
DIMENSION K9FBUF(8)
DATA K9FBUF(8)/" "/
CALL TRACE

C
CALL UNOETS
CALL OETS(K9FBUF(2),36,'0')
IF(FLO(0.30,K9FBUF(2)).NE.'$SADD').AND.
. . (FLO(0.30,K9FBUF(2)).NE.'#SADD') GO TO 890
FLO(0.30,K9FBUF(2)).= BLANK OUT SADD
K9FBUF(1).='SSTART'
CALL ERCSF(NAO,K9FBUF)
GO TO 810

C
C FLAG BAD SADD ELEMENT

C 810 CALL WARNS('BAD SADD FILE OR ELEMENT --')
GO TO 900

C
C ABNORMAL RETURN

C 900 RETURN

C
C NORMAL RETURN

C 900 KOMO=''
RETURN
END
SUBROUTINE KM0AS( 8 BASO -- DYNAMIC BASO
U KOND) & I: FIRST 3 CHARS OF COMMAND O: SPACES

C
C (E M SCHLOSSER)
C
C INCLUDE KMXOT.LIST
CALL TRACE
C
RETURN & NOT YET IMPLEMENTED
END
SUBROUTINE KMOOBR -- DYNAMIC SBRKPT
U KOMOd -- FIRST 3 CHAR OF COMMAND 01 SPACES

(IN H SCHLOSSER)

INCLUDE KOMKGT.LIST
INCLUDE AISHDEF.LIST
DIMENSION KSFBUF(7)
DATA KSFBUF(7) =''
DIMENSION NULASC(I)
DATA NULASC/00001
CALL TRACE

GET SBRKPT IMAGE

CALL UNGETS
CALL GETSKH(KSFBUF,36,-0)
IF(KSFBUF(1).NE.'SBRKPT') GO TO 800
IF(AISH(KSFBUF(2)).NE.'84444') GO TO 800
AISH(KSFBUF(1)) =''

FIX FOR SYNC PROBLEM IN UNIVAC DEMAND SYMBIONT

DO 500 I=1,8
CALL EAPRINT(0,1,NULASC) -- PRINT 4 ASCII NULS
CALL ERTWAT(1000) -- TIMED WAIT 1 SECOND
500 CONTINUE
CALL ERTCN(1,'R ',) -- INSERT LOGICAL BREAK INTO PRINTS FILE
CALL ERTWAT(1000) -- TIMED WAIT 1 SECOND

REQUEST SBRKPT

CALL ERCSF(INAO,KSFBUF)
IF(INAO.NE.0) CALL WARNS('BAD SBRKPT --')
KMOD=''
RETURN

ABNORMAL RETURN

800 RETURN
END
SUBROUTINE KNDOFR - DYNAMIC SFREE
U KOM1 0 1: FIRST 3 CHAR OF COMMAND 0: SPACES

C (E H SCHLOSSER)

INCLUDE KOM01.LIST
CALL TRACE

RETURN 8 NOT YET IMPLEMENTED
END
SUBROUTINE KMOOLO  SLOG -- DYNAMIC SLOG
U KOND) 0 1: FIRST 3 CHARS OF COMMAND 0: SPACES
---------

(E W SCHLOSSER)

INCLUDE KOMXGT.LIST
INCLUDE ASMFAT.LIST
DIMENSION KSFBUF(13)
DATA KSFBUF(13)/' '/
CALL TRACE

CALL UNGETS
CALL GETSKH(KSFBUF,72,0)
IF(FLOD(0,30,KSFBUF(1)),NE.,'SLOG') GO TO 800
ASHSI(KSFBUF(1))=''
CALL ERSBF(NAO,KSFBUF)
IF(NAO.NE.0) CALL WARNS('BAD SLOG -- ')
KOND=''
RETURN

ABNORMAL RETURN

800 RETURN
END
SUBROUTINE KMXXEO: 8 EDIT ACTUAL SPECS INTO MACRO COMMAND DEFINITION

* 1 MACORD: 8 NAME OF MACRO COMMAND
1 HAMFILE: 8 NAME OF FILE
1 NAMELT: 8 NAME OF ELEMENT
1 LOCFILE: 8 LOCATION WITHIN OSF OF MACRO COMMAND DEFINITION

HISTORY

E N SCHLOSSER  LEC  01/17/79  ORIGINAL CODE
E N SCHLOSSER  LEC  05/01/79  FIX MISSPELLING OF HOMFILE AFTER 130

METHOD

GET/CHECK NAME OF MACRO COMMAND AND NUMBER OF FORMAL SPECS FROM DEFINITION IN OSF. COPY REMAINDER OF DEFINITION TO EDIT FILE. REPLACING REFERENCES TO FORMAL SPECS (STRINGS OF '7') WITH ACTUAL SPECS. IF NO DIAGNOSTICS HAVE BEEN ENCOUNTERED, ADD EDIT FILE TO SYSIN RUNSTREAM.

MACHINE-DEPENDENT CODE

UNIVAC EXEC & PROGRAM FILE NAMING CONVENTIONS.
DIMENSION & FORMAT SPECIFICATIONS ASSUME 6 CHARACTERS PER WORD.

EXTERNAL REFERENCES

SYSAUX 8 ADD DISK SYMBOLIC FILE OR ELEMENT TO SYSIN RUNSTREAM
GETS 8 GET REMAINDER OF UNIT S BUFFER
GETSKH 8 GET CHARACTER STRING FIELD FROM UNIT S
HWRMS 8 PRINT/LOG WARNING MESSAGE
SYSGET 8 GET NEXT RECORD FROM SYSIN RUNSTREAM
GETCHR 8 GET CHARACTER FROM CHARACTER STRING
PUTCHR 8 PUT CHARACTER INTO CHARACTER STRING
PUTICE 8 PUT I-C-E INTO CHARACTER STRING
KMXXEO 8 GET/EVALUATE ACTUAL SPEC FOR MACRO COMMAND
GETOKH 8 GET CHARACTER STRING DATA FIELD FROM BUFFER
MDWARN 8 SUBMIT WARNING DIAGNOSTIC MESSAGE
HOFATL 8 SUBMIT FATAL DIAGNOSTIC MESSAGE
MOVCST 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

EXCEPTIONS
1. THE FOLLOWING CONDITIONS GENERATE NOTES:
   - No name (or bad name) declared in macro definition

2. THE FOLLOWING CONDITIONS GENERATE WARNINGS:
   - Macro command nested more than 9 deep
   - Macro definition empty
   - Macro definition longer than 80 characters
   - Formal spec in macro definition does not start with ‘c’
   - More than max0n formal specs declared in macro definition
   - Actual spec missing
   - Actual spec starts with ‘c’
   - Actual spec longer than 12 characters
   - More actual specs than formal specs
   - Undeclared spec referenced in macro definition
   - Macro edit image longer than 80 characters

3. THE FOLLOWING CONDITIONS GENERATE FATAL ERRORS:
   - DBF with macro definition not found
   - Macro definition contains exec command
   - Macro edit file not found

GLOBAL DECLARATIONS

```
#include k00010.List // COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
#include NULL02.List // DEFINE NULL CHARACTER STRING
```

LOCAL DECLARATIONS

```
parameter maxon=6 // maximum number of spec fields supported
integer nom // spec field number (number of question marks)
integer nomfil // number of formal spec fields declared

integer makom(3) // argument
integer macrof(3) // macro command definition field
integer kmari // first character in spec/token
integer nosave // value of nototl on entry to subroutine

integer inoin(20) // definition image input buffer
integer lin(3) // location/length pointers for inoin:
define locin=lin(1) // location of previous/current token in inoin
define lenin=lin(2) // length of previous/current token in inoin
define maxin=lin(3) // length of inoin

integer inout(3) // edited image output buffer
integer lout(3) // location/length pointers for inout:
define locout=lout(1) // location of previous/current token in inout
define lenout=lout(2) // length of previous/current token in inout
define maxout=lout(3) // length of inout

integer imosf(3,makom) // machine dependent
define imosf(i,nom=imosf(i,nom)) // actual spec field image buffer
integer lensf(maxon) // table of length pointers for imosf
```

N-158
PROCEDURE

CALL TRACE

CALL INITIALIZE

GET/CHECK NAME FROM MACRO DEFINITION DFS

GET/CHECK/TALLY FORMAL SPEC(S) FROM MACRO DEFINITION DFS
DAN PACKAGE APPENDIX M

COMMAND ROUTINES

* 'BAD FORMAL SPEC IN MACRO DEFINITION')
  IF(INQM.GT.MAXQ) CALL MOWARN(
  * 'EXTRA FORMAL SPEC IN MACRO DEFINITION')

160 CONTINUE

C GET/CHECK/TALLY ACTUAL SPEC FIELD(S) FROM SYSIN RUNSTREAM

C 200 IF(NQMPL.EQ.0) GO TO 240
  DO 220 NQM=1,NQMPL
     IMOSPFINQM)=NUL'
     CALL KMXXOS(IMOSPFINQM),(18)
     IF(IMOSPFINQM.EQ.'NUL') CALL MOWARN('MISSING SPEC')
     CALL GETCHR(KHARI,IMOSPFINQM),(111)
     IF(KHARI.EQ.'(') CALL MOWARN('BAD SPECIFICATION --')
     LENSPF(NQM)=LENCHIMOSPFINQM),(18)
     IF(LENSPF(NQM).GT.12) CALL MOWARN(
        * 'SPECIFICATION TOO LONG --')
     220 CONTINUE

C FLUSH EXTRA ACTUAL SPECIFICATION(S) FROM SYSIN RUNSTREAM

C 240 ITEM='NUL'
     CALL GETS$K(ITEM,(4),NULCST)
     IF(ITEM.EQ.'NUL') CALL MOWARN('EXTRA SPECIFICATION --')
     240 CONTINUE

C IF DIAGNOSTICS ENCOUNTERED FLUSH ADD-ED MACRO DEFINITION

C 420 CALL SYSOET(INSTAT,1NQM,MAXIN)
     IF(INSTAT.EQ.' ') GO TO 420
     IF(INSTAT.NE.'EOA') CALL MOWARN('EXEC COMMAND IN MACRO DEFINITION')
     500 CONTINUE

C READ NEXT DEFINITION IMAGE

C 500 IF(LOCIN.NEQ.1) MAXIN=MAXIN+5
     IF(LOCIN.EQ.1) MAXIN=(ABS(MAXIN))
     IF(LOCIN.LE.0) CALL SYSOET(INSTAT,1NQM,MAXIN)
     IF(INSTAT.EQ.' ') GO TO 600
     IF(INSTAT.NE.'EOA') GO TO 600
     IF(MAXIN.LE.0) CALL MOWARN('MACRO DEFINITION IMAGE TOO LONG')
     MAXIN=M0N(MAXIN,80)

C INITIALIZE EDIT IMAGE WITH BLANKS

C LOCOUT=1
     CALL HVCST(IMOOUT,(LOCOUT),(84',',',',',',)',',',',')

C LOCATE NEXT DEFINITION TOKEN
V5

DAN PACKAGE APPENDIX M

COMMAND ROUTINES

520 CALL NEXTOK(KHAR1.LLIN., LLIN.1MODIN, ".?", 'NONE')
  IF(MAXIN.LE.0) GO TO 580 ELSE OF IMAGE
  IF(KHAR1.EQ.'?') GO TO 530

APPEND NON-SPEC TOKEN TO EDIT IMAGE

LENOUT=LENIN
  IF(LENOUT.LE.(81-LOCOUT)) CALL MOVCST(IMGOUT.(LOCOUT),(LENOUT),
  IMGIN.(LOCIN),(LENIN), '*')
  LOCOUT=LOCOUT+LENOUT
  GO TO 520

APPEND ACTUAL SPEC TO EDIT IMAGE IN PLACE OF FORMAL SPEC TOKEN

530 IF(LININ.GT.NOMFML) CALL MDWARN( 
  'UNDECLARED SPEC REFERENCED IN MACRO DEFINITION')
  NOM=NINO(LENIN,NOMFML)
  LENOUT=LENSPF(NOM)
  IF(LENOUT.LE.(81-LOCOUT)) CALL MOVCST(IMGOUT.(LOCOUT),(LENOUT),
  IMGSF(NOM),(1),(LENOUT), '*')
  LOCOUT=LOCOUT+LENOUT
  GO TO 520

WRITE EDIT IMAGE

560 MAXOUT=LOCOUT-1
  IF(MAXOUT.GT.80) CALL MDWARN( 'MACRO EDIT LINE TOO LONG')
  IF(NOSAVE.EQ.NOTOTLI WRITE(LUEOIT,575) IMGOUT
  575 FORMAT(14A6) MACHINE DEPENDENT FORMAT
  GO TO 500

DRAIN READS BUFFER & APPEND ITS CONTENTS (IF ANY) TO EDIT FILE

600 IMGOUT(1)=' NUL'
  CALL GETS(IMGOUT,(84), NULCST)
  IF(IMGOUT(I).NE.' NUL') WRITE(LUEOIT,615) IMGOUT
  615 FORMAT(14A6)

CLOSE EDIT FILE AND ADD TO SYSIN RUNSTREAM

IF(INSTAT.NE.'EOA') CALL MDFATL(' 
  'EXEC COMMAND IN MACRO DEFINITION')
ENDFILE LUEDIT
  CALL CLOSE(LUEDIT,1) A CLOSE & REWIND
  IF(NOSAVE.NE.NOTOTLI GO TO 900
  NFEDIT(1)=
  NFEDIT(2)=
  CALL PUTICE(NFEDIT,(1), ICE(3)'-LUEDIT/10) A TENS DIGIT
  CALL PUTICE(NFEDIT,(1), ICE(3)'-LUEDIT/10*(LUEDIT/10)) A UNITS DIGIT
  CALL SYSAD(LOCFIL, NFEDIT, ' ', ' ')

M-155
IF(LOCFIL.LT.0) CALL MDFATL('NO MACRO EDIT FILE IN KMXXED')

C CONFIRM MACRO COMMAND & SPECS
C
IF(MCFIRM.EQ.0) GO TO 900
CALL MOV CST(IMGIN,(1),(120),
    MAKOMD,(1),(12),'',
    LOCNXT+LEN CST(IMGIN,120)=1
IF(NMFML.LT.0) GO TO 730
DO 710 NOM=1,NMFML
   IF(LOCNXT.GT.105) GO TO 730
   CALL PUTCRT(IMGIN,(LOCNXT),',',
   LOCNXT=LOCNXT+2
   CALL MOV CST(IMGIN,(LOCNXT),(12),
      IMOSPF(NOM),(1),(12),',',
   LOCNXT=LOCNXT+LEN SPF(NOM)
   CONTINUE
710  WRITE(G,735) IMGIN
735 FORMAT(IX,20A6)
C
C DONE
C
900 RETURN
END
SUBROUTINE KMXXOS -- GET/EVALUATE ACTUAL SPEC FOR MACRO COMMAND
O CMPLD. 8 ACTUAL SPEC FIELD (UNCHANGED IF MISSING)
( KMLEN 8 LENGTH IN CHARMS (WILL BE PADDED WITH BLANKS TO WORD ROY)

C---------------------------------------------------------------------
C HISTORY
C-------
C
C METH00
C-----
C
C	 GET ACTUAL SPECIFICATION (IF ANY). IF ITS AN EXPRESSION (ENCLOSED
C	 IN PARENTHESES) CHECK IT. EVALUATE IT. AND RETURN ITS VALUE.
C
MACHINE-DEPENDENT CODE
---------------------------------------
NONE.

EXTERNAL REFERENCES
-------------------
GETSKH   8 GET CHARACTER STRING FIELD FROM UNIT 5
MOVCST   8 MOVE CHARACTER STRING
INTEGER ICE 8 INTEGER-CHARACTER-EQUIVALENT
INTEGER LCREQ 8 LOCATE CHARACTER IN STRING EQUAL TO SEARCH CHARACTER
INTEGER LENPAD 8 LENGTH IN CHARACTERS INCL PAD TO WORD BOUNDARY

EXCEPTIONS
----------
1. AN INVALID EXPRESSION IS RETURNED WITH THE LEADING '1' CHANGED
   TO 'C'.

GLOBAL DECLARATIONS
------------------
INCLUDE KOMXQT.LIST 8 COMMON PROGRAM EXECUTION SWITCHES, COUNTERS
INCLUDE KOMKLS.LIST 8 COMMON CLASSIFICATION INFO
INCLUDE NULCST.LIST 8 DEFINE NULL CHARACTER STRING

LOCAL DECLARATIONS
--------------------
INTEGER KMEXP 8 EXPRESSION
INTEGER MODESET 8 MODE SETTING OF SOME MODE SWITCH
CALL TRACE

CALL GET ACTUAL SPECIFICATION FIELD

CALL GETSKMH(KHFLO, (KMLEN), NULCST)
IF (ICE(KHFLO).NE.ICE('')). GO TO 900  -- NOT AN EXPRESSION

EXTRACT 1ST 3 CHAR OF EXPRESSION FROM ENCLOSING PARENTHESES

IF (LCRREG(KHFLO, (2), (13), ('')).NE.0) GO TO 240
CALL PUTCHR(KHFLO, (1), '"')  -- FLAG AS INVALID EXPRESSION
GO TO 900
240 CALL MOVCST(KHEXP, (1), (LENPAD(3)),
  "KHFLO, (2), (3), '"')

CHECK/EVALUATE GENERAL EXPRESSION

IF (KHEXP.NE. 'DET') GO TO 310  -- DETECT IT?
   CALL MOVCST(KHFLO, (1), (LENPAD(KMLEN)),
   "KLSTYP, (1), (3), '"')
GO TO 900
310 CONTINUE  -- FUTURE CODE

CHECK/EVALUATE MODE SWITCH EXPRESSION

500 IF (KHEXP.NE. 'BAT') GO TO 510  -- BAT(Ch)?
  MODSET=MBATCH
  GO TO 700
510 IF (KHEXP.NE. 'CHE') GO TO 520  -- CHeCKOUT?
  MODSET=MCHECK
  GO TO 700
520 IF (KHEXP.NE. 'CON') GO TO 530  -- CON(FIRM)?
  MODSET=MCONFIRM
  GO TO 700
530 IF (KHEXP.NE. 'DUM') GO TO 540  -- DUMP?
  MODSET=MDUMP
  GO TO 700
540 IF (KHEXP.NE. 'ECH') GO TO 550  -- ECH(0)?
  MODSET=MECHO
  GO TO 700
550 IF (KHEXP.NE. 'LEO') GO TO 560  -- LE(End)?
  MODSET=MLEEND
  GO TO 700
560 IF (KHEXP.NE. 'PRO') GO TO 570  -- PROMPT?
  MODSET=MPROMT
  GO TO 700

M-158
570 IF(KHEXP.NE.'TRA') GO TO 580  & TRAICE1?
MODSET=IMTRACE
GO TO 700
580 CONTINUE  & FUTURE CODE
C
C INVALID EXPRESSION -- PUT ' ' IN FIRST CHARACTER
C
CALL PUTCHR(KMFLO.(1), ' ')  
GO TO 900
C
C STORE EVALUATION OF MODE SWITCH EXPRESSION
C
700 KHEXP='ON'
IF(MODSET.EQ.0) KHEXP='OFF'
CALL MOVCST(KMFLO.(1),(LENPAD(KHLEN)).
     ' ', KHEXP.(1),(3),'' )
C
C DONE
C
900 RETURN
END

C
APPENDIX N
UTILITY ROUTINES

SPT.SC DAM.PREFACE-N
SPT.SC DAM.APPENDIX-N
SPT.SC DAM.RESOURCE
SPT.SC DAM.ATRACE/DAM
SPT.SC DAM.CALWIN
SPT.SC DAM.CLOSE3
SPT.SC DAM.CLOSE4
SPT.SC DAM.DSSPR
SPT.SC DAM.DSSPR3
SPT.SC DAM.DZDNS
SPT.SC DAM.EISRTD
SPT.SC DAM.ENVOR1
SPT.SC DAM.ENWIN
SPT.SC DAM.EOF
SPT.SC DAM.FACPR
SPT.SC DAM.FINFO
SPT.SC DAM.OCRER
SPT.SC DAM.OCHOM
SPT.SC DAM.OCFF
SPT.SC DAM.OCINF
SPT.SC DAM.OETOSR
SPT.SC DAM.GETRAD
SPT.SC DAM.GET5
SPT.SC DAM.GETSAL
SPT.SC DAM.GETSFR
SPT.SC DAM.GET5IN
SPT.SC DAM.GET5KH
SPT.SC DAM.GET5RL

(8009)

SET TABS 8 12 & 31

DUMP A-REGISTER IN OCTAL (FOR DEBUGGING)
DETERMINE # OF ACTUAL ARGUMENTS & RETURN VECTOR
TRACING CALLS TO ASSEMBLER ROUTINES
BOX CHARACTERS
EXTRACT/SCALE/DUMP BYTE FIELDS (2'S COMPL)
CALCULATE CHANNEL POINTERS
CALCULATE COLOR/INTENSITY IN SYMBOL TABLE
CALCULATE PRINT/PILOT COEFFICIENTS FOR SCALE
CALCULATE PRINT/PILOT COEFFICIENTS FOR SPACING
CALCULATE SYMBOL TABLE FORPRINTING
CALCULATE WINDOW ENVELOPES
CLOSE UNIT 3 (INPUT ERTS MSS DATA)
CLOSE COMMAND RECALL FILE (UNIT 4)
CLOSE AND PRINT ALTERNATE PRINT (SPOOL) FILES
WRITE COMMON ID/CLASSIFICATION HEADING
CLOSE UNIT 3 (INPUT ERTS MSS DATA IN BIP FORMAT)
CORRELATIONS/MEANS/DEVS FROM SUMS/SUMS-OF-PRODUCTS
CROP MSS OUTPUT WINDOW TO FIT PRINT FILE SIZE
CORRELATIONS/MEANS/DEVS FROM SUMS/SUMS-OF-PRODUCTS
CONVERT DEGREES.MINUTES.SECONDS TO DEGREES
DELETE ALL DENSITY FILES
DBL PRECISION GAUSS-JORDAN REDUCTION
COMPUTE X OF TRACE FOR MATRIX DIAGONAL ELEMENTS
RECURSIVE OF MATRIX DIAGONAL ELEMENTS
SQUARE ROOT OF MATRIX DIAGONAL ELEMENTS
DELETE ALTERNATE PRINT FILES (SEE CLOSPR)
DELETE INPUT DETECTION FILES (SEE OPM12N)
DUMP TYPICAL PROGRAM
DUMP WINDOW PACKET
COMPUTE DBL PRECISION SUMS & SUMS-OF-PRODUCTS
DBL PREC SUMS & SUMS-OF-PRODUCTS FROM UNIT 3
DEGREES TO DEGREES, MINUTES, SECONDS
SORT E-VALUES/E-VECTORS BY DESCENDING E-VALUES
ADD ORIGIN TO ENVELOPE (REAL WINDOW PACKET)
COMPUTE ENVELOPE FOR REAL WINDOW PACKET
END-OF-FILE (ECF)
PRINT FACTOR STRUCTURE/COEFFICIENTS/MEANS
GET FILE DESCRIPTIVE INFORMATION
LOAD 'ERT' GEOMETRIC CONSTANTS
LOAD 'HOM' GEOMETRIC CONSTANTS
LOAD 'LCC' GEOMETRIC CONSTANTS (SEE OCHOM)
LOAD 'LCC' GEOMETRIC CONSTANTS (SEE OCHOM)
LOAD PROPER GEOMETRIC CONSTANTS
LOAD 'PS' GEOMETRIC CONSTANTS (SEE OCHOM)
LOAD 'SOM' GEOMETRIC CONSTANTS (SEE OCHOM)
LOAD 'UTM' GEOMETRIC CONSTANTS (SEE OCHOM)
GENERATE/STORE/LIST-SORT 'TICKS'
GET DISK SYMBOLIC RECORD
GET ORIGINAL/TRANSFORMED RADIANCE FROM ERTS TAPE
GET REMAINING CONTENTS OF UNIT 5 BUFFER
GET ALPHA DATA FIELD FROM UNIT 5 (SEE GET5)
GET FRACTION DATA FIELD FROM UNIT 5 (SEE GET5)
GET INTEGER DATA FIELD FROM UNIT 5 (SEE GET5)
GET CHARACTER DATA FIELD FROM UNIT 5 (SEE GET5)
GET REAL DATA FIELD FROM UNIT 5 (SEE GET5)
APPENDIX-N

UTILITY ROUTINES

- GET SEX-Y DATA FIELD FROM UNIT 5 (SEE GETS)
- PRINT HEADING LINE(S)
- ALLOCATE ARRAY OF 4390 WORDS IN 1-BANK
- PRINT SHORT ERTS SCENE IDENTIFICATION
- PRINT COMPLETE ERTS SCENE IDENTIFICATION
- PRINT SHORT ID FOR LOGICAL UNIT 3
- INTEGER DUPLICATE (INDIRECT REF TO OPTIONAL ARG)
- GET FORTRAN I/O STATUS (UNIVAL SYSTEM ROUTINE)
- ADD ORIGIN TO ENVELOPE (INTEGER WINDOW PACKET)
- COMPUTE ENVELOPE FOR INTEGER WINDOW PACKET
- INTEGER-COLOR-EQUIVALENT FOR COLOR
- JACOBI ITERATION TO FIND EIGEN-VALUES/VECTORS
- JOIN BUFFERS FROM TWO DETECTION FILES
- COLOR FOR INTEGER-COLOR-EQUIVALENT
- SPREAD COUNT FLAGS INTO INTERIOR UNDEFINED PIXLS
- LINE OF BOX DIGIT FOR INTEGER
- LOAD NOMINAL REGISTRATION PARAMETERS (SEE LOREOB)
- LOAD EXACT REGISTRATION PARAMETERS FROM UNIT 8
- LOCATE DISK SYMBOLIC FILE OR ELEMENT
- LOGARITHM, BASE 2 (TRUNCATED) OF INTEGER
- WRITE MAP WINDOW HEADING
- PRINT MATRIX
- CLEAR 'FATAL ERROR' COUNT (SEE MOLOO)
- CLEAR 'WARNING' COUNT (SEE MOLOO)
- PRINT/LOG/COUNT 'FATAL ERROR' (SEE MOLOO)
- LOG DIAGNOSTIC MESSAGES
- PRINT/LOG/NOTE' (SEE MOLOO)
- PRINT/LOG/COUNT 'WARNING' (SEE MOLOO)
- MASK PIXELS IN BUFFER OUTSIDE NON-TRIVIAL WINDOW
- MOVE CONTENTS BETWEEN SPECIFIED LOCATIONS
- MATRIX MULTIPLICATION
- CONVERT DIGITAL PICTURE FROM POSITIVE TO NEGATIVE
- I/O UNIT NUMBER TABLE
- DO ABSOLUTELY NOTHING!
- NAME 'VIA' 'TO' ROUTINES
- OPEN ALTERNATE PRINT (SPOOL) FILES (UNITS 10-19)
- OPEN UNIT 3 (INPUT ERTS MSS DATA)
- OPEN COMMAND RECALL FILE (SEE WRITE)
- OPEN INPUT DETECTION FILES (UNITS 21-24)
- OPEN UNIT 3 (INPUT DATA ON DISK IN PXBDEF FMT)
- OPEN UNIT 3 (INPUT ERTS MSS DATA IN MOP FORMAT)
- OPEN UNIT 3 (INPUT ERTS MSS DATA ON TAPE)
- OPEN UNIT 3 (MOP FORMAT ANCILLARY RECORDS)
- OPEN UNIT 3 (MOP FORMAT ANNOTATION RECORDS)
- OPEN UNIT 3 (MOP FORMAT HEADER RECORD)
- OPEN UNIT 3 (MOP FORMAT SIZE & INPUT WINDOW)
- OPEN UNIT 3 (MOP FORMAT SIZE & INPUT WINDOW)
- OPEN UNIT 3 (MOP FORMAT SIZE & INPUT WINDOW)
- OPEN UNIT 3 (MOP FORMAT SIZE & INPUT WINDOW)
- OPEN UNIT 3 (MOP FORMAT TAPE DIRECTORY RECORD)
- PROGRAM ABORT (SEE PSTOP)
- ESTIMATE PITCH AND ROLL
- PRINT BOX NUMBERS (VARIABLE HEIGHT)
- PRINT/OVERPRINT FILES
- PRINT/OVERPRINT SYMBOL BUFFER
APPENDIX N
UTILITY ROUTINES

APPENDIX N

- PRINT SYMBOL LEGEND
- WRITE BOX CHARACTERS ON ANY UNIT
- SET PRINT LINES PER INCH
- SET PRINT MARGINS & LINES PER PAGE
- PROGRAM INITIATION
- PROGRAM TERMINATION
- DUMP PXBOEF PIXEL BUFFER PREAMBLE FOR DEBUGGING
- PXBOEF PREAMBLE FOR MOP 'AM' PREAMBLE
- PXBOEF PREAMBLE FOR MOP 'AR' PREAMBLE
- PXBOEF PREAMBLE FOR MOP 'PH' PREAMBLE
- PXBOEF PREAMBLE FOR MOP 'PR' PREAMBLE
- FIT Y = A*X**2 + B*X + C & SOLVE FOR EXTREMA
- NORMALIZED QUARTIMAX ROTATION CRITERION
- UN-NORMALIZED QUARTIMAX ROTATION CRITERION
- READ UNIT 3 (ERTS MSS DATA IN MOP BIL FORMAT)
- READ UNIT 3 (ERTS MSS DATA IN BIP FORMAT)
- READ UNIT 3 (ERTS MSS DATA IN MOP BSQ FORMAT)
- READ UNIT 3 (DATA ON DISK IN PXBOEF FORMAT)
- READ UNIT 3 (SYNTHETIC DATA WHEN NO UNIT 3)
- READ DATA FROM DETECTION FILE(S) (UNITS 21-24)
- READ UNIT 3 (ERTS MSS DATA)
- FILL BUFFER FOR UNIT 5 (CARD READER OR TERMINAL)
- NOMINAL REGISTRATION PARAMETERS
- NOMINAL REGISTRATION PARAMETERS
- NOMINAL REGISTRATION PARAMETERS
- NOMINAL REGISTRATION PARAMETERS
- NOMINAL REGISTRATION PARAMETERS
- NOMINAL REGISTRATION PARAMETERS
- NOMINAL REGISTRATION PARAMETERS
- NOMINAL REGISTRATION PARAMETERS
- REVERT EQUATIONS
- WRITE/ADD SPECIFIED SYMBOLIC ELEMENT
- CONVERT REAL TO SEXAGENARY ARRAY (INTEGER)
- CONVERT REAL TO SEXAGENARY ARRAY (REAL)
- COMPUTE REAL FROM SEXAGENARY ARRAY (REAL)
- ROTATE TWO MATRIX COLUMNS TO MAXIMIZE FUNCTION
- ROTATE TWO MATRIX COLUMNS
- ROTATE TWO MATRIX ROWS
- READ UNIT 3 (READ ONE RECORD FROM TAPE)
- GET/SET MODE SWITCHES
- SHARPEN SAMPLES IN PXBOEF FORMAT BUFFER
- SHIFIT BITS CIRCULAR WITHIN WORDS OF ARRAY
- ENABLE/DISABLE SPANNING FOR UNIT 5
- SPLIT REAL INTO SIGN, INTEGER, DECIMAL
- SPANNED READ OF UNIT 5 (USED ONLY BY GETS)
- COMPUTE SUMS & SUMS-OF-PRODUCTS
- STORE REGISTRATION PARAMETERS ON UNIT 5
- GENERATE SUBWINDOW MAPS
- TRACE CALLS TO FORTRAN ROUTINES
- TAPE ERROR RECOVERY FOR UNIT 3 (MOP FORMAT TAPES)
- TAPE SNAP FOR UNIT 3 (MOP FORMAT TAPES)
- BACK UP 1 DATA FIELD ON UNIT 5 INPUT (SEE GETS)
- VALIDATE SECURITY KEY
- NORMALIZED VARIMAX ROTATION CRITERION
- UN-NORMALIZED VARIMAX ROTATION CRITERION
DAM PACKAGE APPENDIX N
UTILITY ROUTINES

SPRT.SC DAM.VER40
SPRT.SC DAM.VER44P
SPRT.SC DAM.VER43U
MSG,N   .VIATO
SPRT.SC DAM.WARN5
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)
<table>
<thead>
<tr>
<th>SPRT.SC DAM.PREFACE-0</th>
<th>0009</th>
<th>SET Tabs 8 &amp; 21</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPRT.SC DAM.APPENDIX-0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAM.A40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAM.A4P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAM.O4U/CLARKE1866</td>
<td></td>
<td>ADJUSTED MSS COORDINATES FOR GEOGRAPHIC COORD</td>
</tr>
<tr>
<td>SPRT.SC DAM.O4A</td>
<td></td>
<td>GEOGRAPHIC COORDINATES FOR ADJUSTED MSS COORD</td>
</tr>
<tr>
<td>SPRT.SC DAM.O4P</td>
<td></td>
<td>GEOGRAPHIC COORDINATES FOR PRT/PLT COORD</td>
</tr>
<tr>
<td>SPRT.SC DAM.O4U/CLARKE1866</td>
<td></td>
<td>UTM COORDINATES FOR GEOGRAPHIC COORDINATES</td>
</tr>
<tr>
<td>SPRT.SC DAM.P4A</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAM.P40</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SPRT.SC DAM.O4U/CLARKE1866</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX-P

EXECUTIVE REQUESTS

00000

10005

PRINT IMAGE ON TTY OR LINE PRINTER (ASCII)

PRINT IMAGE FROM TTY OR CARD READER (ASCII)

SUBMIT EXEC COMMANDS (FIELDATA)

READ IMAGE FROM TTY OR CARD READER (FIELDATA)

RETURN SYSTEM DATE AND TIME (FIELDATA)

ENDS TERMINATE PROGRAM

TERMINATE PROGRAM IMMEDIATELY

RETIEVE FACILITIES ASSIGNMENT INFORMATION

RETIEVE FACILITIES ASSIGNMENT INFORMATION

RETIEVE SYSTEM/RUN/PROGRAM/FILE INFO

INITIATE I/O

INITIATE I/O & WAIT FOR COMPLETION

WRITE IMAGE TO ALTERNATE PUNCH FILE (FIELDATA)

RETRIEVE PART OF PROGRAM CONTROL TABLE (PCT)

PROGRAM FILE SEARCH FOR INFO ON ELEMENT

SET ALTERNATE PRINT FILE CONTROLS (FIELDATA)

SET PRINT FILE CONTROLS (FIELDATA)

WRITE IMAGE TO ALTERNATE PRINT FILE (FIELDATA)

READ IMAGE FROM TTY OR CARD READER (FIELDATA)

READ IMAGE FROM READ ALT FILE (FIELDATA)

READ IMAGE FROM READ ALT FILE (FIELDATA)

RETURN ACCUMULATED SUPS (200 USEC INCR) FROM PCT

SUBMIT EXEC REQUESTS (UNIVAC SYSTEM ROUTINE)

SNAP TAPE REELS OF MULTI-REEL FILE

SNAP TAPE REELS OF MULTI-REEL FILE

TIME WAIT UP TO 30 SECONDS

WAIT FOR COMPLETION OF I/O

SUBMIT EXEC REQUESTS (UNIVAC SYSTEM ROUTINE)

ADD DISK SYMBOLIC FILE OR DLT TO SYSSIN RUNSTREAM

GET NEXT RECORD FROM SYSSIN RUNSTREAM

OUTPUT RECORD TO SYSSOUT PRIMARY CRT/PRINTER FILE
### MACROS

- **SPRT.SC** DAM.KOMKOT-PROC: COMMON PROGRAM EXECUTION INFO (ASM ROUTINES ONLY)
- **SPRT.SC** DAM.KOMK-PROC: NAMELIST SPECS FOR UNIT 8 INTEGRATION PARAMETERS
- **SPRT.SC** DAM.KOMK-PROC: DEFINE MAXIMUM BYTE VALUE
- **SPRT.SC** DAM.KOMK-PROC: DEFINE MAXIMUM INTEGER-CHAR-EQIY VALUE
- **SPRT.SC** DAM.KOMK-PROC: DEFINE MAXIMUM INTEGER VALUE
- **SPRT.SC** DAM.KOMK-PROC: COMMON/HEADER BLOCKS FOR ERTS DETECTION FILES
- **OMSG.N** .NITAB: (SEE ALT.PRT-PROCS)
- **OMSG.N** .NITROT: (SEE ALT.PRT-PROCS)
- **SPRT.SC** DAM.KOMK-PROC: DEFINE NULL CHARACTER
- **SPRT.SC** DAM.KOMK-PROC: DEFINE NULL CHARACTER STRING
- **SPRT.SC** DAM.KOMK-PROC: DEFINE PICTAB PARAMETERS
- **OMSG.N** .MTPDNC: PPD LINE FOR CORRECTED (SEE TRFORM-PROCS)
- **OMSG.N** .PPDCNC: PPD COLUMN FOR CORRECTED (SEE TRFORM-PROCS)
- **SPRT.SC** DAM.KOMK-PROC: DEFINE PRCLASS PARAMETERS
- **SPRT.SC** DAM.KOMK-PROC: DEFINE PRDEF PARAMETERS
- **SPRT.SC** DAM.KOMK-PROC: DEFINE STRUCTURE OF PIXEL BUFFER
- **OMSG.N** .STMN4C: 8TH NORTHING FOR CORRECTED (SEE TRFORM-PROCS)
- **OMSG.N** .STNC4C: 8TH EASTING FOR CORRECTED (SEE TRFORM-PROCS)
- **OMSG.N** .STX4QT: (SEE XQTLOG-PROCS)
- **SPRT.SC** DAM.KOMK-PROC: DEFINE COORDINATE TRANSFORMATIONS
- **OMSG.N** .XINDEF: (SEE WINDOW-PROCS)
- **SPRT.SC** DAM.KOMK-PROC: COMMON/DEFINE FOR WINDOW Packets
- **OMSG.N** .XQTDEFS: (SEE XQTLOG-PROCS)
- **SPRT.SC** DAM.KOMK-PROC: COMMON BLOCKS FOR XQT & DAM LOG MANIPULATION
### CHAR/BYTE/STRING ROUTINES

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII BYTE STRING FOR CHARACTER STRING</td>
<td>OAM.CBINIT</td>
</tr>
<tr>
<td>ASCII BYTE STRING FOR EBCDIC BYTE STRING</td>
<td>OAM.CBWRITE</td>
</tr>
<tr>
<td>INTERNAL STRING FOR 8-BIT EXTERNAL STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>INTERNAL STRING FOR 8-BIT EXTERNAL STRING</td>
<td>OAM.GETCHRE</td>
</tr>
<tr>
<td>8-BIT EXTERNAL STRING FOR INTERNAL STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>8-BIT EXTERNAL STRING FOR INTERNAL STRING</td>
<td>OAM.GETCHRE</td>
</tr>
<tr>
<td>INITIALIZE CHARACTER BUFFER</td>
<td>OAM.CBINIT</td>
</tr>
<tr>
<td>CHARACTER BUFFER FOR CHARACTER STRING</td>
<td>OAM.CBWRITE</td>
</tr>
<tr>
<td>CHARACTER BUFFER FOR FILE (&quot;READ&quot;)</td>
<td>OAM.CBFILE</td>
</tr>
<tr>
<td>CHARACTER BUFFER FOR INTEGER</td>
<td>OAM.CBINT</td>
</tr>
<tr>
<td>CHARACTER BUFFER FOR REAL</td>
<td>OAM.CBREAL</td>
</tr>
<tr>
<td>CLEAR QUARTER-WORD MODE</td>
<td>OAM.CBQWD</td>
</tr>
<tr>
<td>CHARACTER STRING FOR ASCII BYTE STRING</td>
<td>OAM.CBST4AS</td>
</tr>
<tr>
<td>CHARACTER STRING FOR EBCDIC BYTE STRING</td>
<td>OAM.CBST4EB</td>
</tr>
<tr>
<td>CHARACTER STRING FOR INTEGER</td>
<td>OAM.CBST4IN</td>
</tr>
<tr>
<td>CHARACTER STRING FOR REAL</td>
<td>OAM.CBST4RL</td>
</tr>
<tr>
<td>CURVATURE (1ST DERIVATIVE) OF BYTE STRING</td>
<td>OAM.CURBST</td>
</tr>
<tr>
<td>CURVATURE (1ST DERIVATIVE) OF CHARACTER STRING</td>
<td>OAM.CURBST</td>
</tr>
<tr>
<td>VARIABLE-LENGTH (&lt; 8 CHAR) STRING FOR CHAR STRING</td>
<td>OAM.CBS4CS</td>
</tr>
<tr>
<td>VARIABLE-LENGTH (&lt; 8 CHAR) STRING FOR INTEGER</td>
<td>OAM.CBS4IN</td>
</tr>
<tr>
<td>VARIABLE-LENGTH (&lt; 8 CHAR) STRING FOR REAL</td>
<td>OAM.CBS4RL</td>
</tr>
<tr>
<td>DECODE NUMERIC CHARACTER STRING</td>
<td>OAM.CDECODE</td>
</tr>
<tr>
<td>EBCDIC BYTE STRING FOR ASCII BYTE STRING</td>
<td>OAM.CST4AS</td>
</tr>
<tr>
<td>EBCDIC BYTE STRING FOR CHARACTER STRING</td>
<td>OAM.CST4EB</td>
</tr>
<tr>
<td>FILE FOR CHARACTER BUFFER (&quot;WRITE&quot;)</td>
<td>OAM.CFILECB</td>
</tr>
<tr>
<td>GET NON-NEG INTEGER FROM BYTE IN BYTE STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>GET CHARACTER FROM CHARACTER STRING</td>
<td>OAM.GETCHRE</td>
</tr>
<tr>
<td>GET NON-NEG INTEGER FROM DOUBLE BYTE IN BYTE STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>GET HEXADECIMAL CHAR FROM NYBLE IN BYTE STRING</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>GET INTEGER-CHAR-EQUIV FROM CHAR STRING (SEE GETCHRE)</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>GET INTEGER FROM INTEGER STRING</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>BEGIN ERROR WALKBACK (ARGS MATCH GETBYTE/CHR/INT)</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>GET NON-NEG INTEGER FROM NYBLE IN BYTE STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>GET INTEGER FROM QUADRUPLE BYTE IN BYTE STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>GET NEXT CHAR STRING DATA (OF BYTE STRING ARRAY)</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>GET ARRAY OF NON-NEG INTEGERS FROM BYTE STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>INTEGER CHARACTER EQUIVALENT (FROM CHARACTER)</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>INTEGER FROM CHARACTER CHARACTER EQUIVALENT</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>INTEGER FOR ONE'S COMPLEMENT</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>INTEGER FOR TWO'S COMPLEMENT</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>* ENCODE 8-CHAR (FIELD DATA) STRING FROM INTEGER</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>ONE'S COMPLEMENT FOR INTEGER</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>TWO'S COMPLEMENT FOR INTEGER</td>
<td>OAM.GETHEX</td>
</tr>
<tr>
<td>LOCATION OF BYTE IN STRING EQUAL TO SEARCH BYTE</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LOCATION OF BYTE IN STRING NOT EQ TO SEARCH BYTE</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LOCATION OF CHAR IN STRING EQUAL TO SEARCH CHAR</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LOCATION OF CHAR IN STRING NOT EQ TO SEARCH CHAR</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LENGTH CHARACTER STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LOCATION IN ONE CHARACTER STRING OF ANOTHER</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LENGTH OF CHARACTER STRING</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LENGTH Padded TO NEXT WORD BOUNDARY</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LOCATION OF ICE IN STRING EQUAL TO SEARCH ICE</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LOCATION OF ICE IN STRING NOT EQ TO SEARCH ICE</td>
<td>OAM.GETBYTE</td>
</tr>
<tr>
<td>LOCATION OF INTEGER IN STRING EQ TO SEARCH INT</td>
<td>OAM.GETBYTE</td>
</tr>
</tbody>
</table>
CHAR/BYTE/STRING ROUTINES

APPENDIX #R

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;DAM.LINTHE</td>
<td>LOCATION OF INTEGER IN STRING NE TO SEARCH INT</td>
</tr>
<tr>
<td>&amp;DAM.LOWCST</td>
<td># OF CHAR STRING LOWER IN COLLATING SEQUENCE</td>
</tr>
<tr>
<td>&amp;DAM.MOVBST/ASH</td>
<td>MOVE BYTE STRING</td>
</tr>
<tr>
<td>&amp;DAM.MOVBST/FORE</td>
<td>MOVE BYTE STRING</td>
</tr>
<tr>
<td>&amp;DAM.MOVBYT</td>
<td>MOVE BYTE</td>
</tr>
<tr>
<td>&amp;DAM.MOVCHR</td>
<td>MOVE CHARACTER</td>
</tr>
<tr>
<td>&amp;DAM.MOVCHT</td>
<td>MOVE CHARACTER STRING</td>
</tr>
<tr>
<td>&amp;DAM.MOVCHT/ASH</td>
<td>MOVE CHARACTER STRING</td>
</tr>
<tr>
<td>&amp;DAM.MOVCHT/FORE</td>
<td>MOVE DOUBLE BYTE</td>
</tr>
<tr>
<td>&amp;DAM.MOVIST</td>
<td>MOVE INTEGER STRING</td>
</tr>
<tr>
<td>&amp;DAM.NBYN1</td>
<td>NUMBER OF BYTES FOR NUMBER OF INTEGERS</td>
</tr>
<tr>
<td>&amp;DAM.NCHN1</td>
<td>NUMBER OF CHARACTERS FOR NUMBER OF INTEGERS</td>
</tr>
<tr>
<td>&amp;DAM.NEXTOK</td>
<td>GET POINTERS TO NEXT TOKEN IN IMAGE BUFFER</td>
</tr>
<tr>
<td>&amp;DAM.NI4NB</td>
<td>NUMBER OF INTEGERS FOR NUMBER OF BYTES</td>
</tr>
<tr>
<td>&amp;DAM.NI4NC</td>
<td>NUMBER OF INTEGERS FOR NUMBER OF CHARACTERS</td>
</tr>
<tr>
<td>&amp;DAM.PUTBYT</td>
<td>PUT NON-NEG INTEGER INTO BYTE OF BYTE STRING</td>
</tr>
<tr>
<td>&amp;DAM.PUTCHR</td>
<td>PUT CHARACTER INTO CHARACTER STRING</td>
</tr>
<tr>
<td>&amp;DAM.PUTDHY</td>
<td>PUT NON-NEG INTEGER INTO DOUBLE BYTE OF BYTE STRING</td>
</tr>
<tr>
<td>&amp;DAM.PUTHEX</td>
<td>PUT HEXADECIMAL CHAR INTO NYBLE IN BYTE STRING</td>
</tr>
<tr>
<td>&amp;DAM.PUTICE</td>
<td>PUT INTEGER-CHAR-EQUIV INTO CHAR STRING (SEE PUTCHR</td>
</tr>
<tr>
<td>&amp;DAM.PUTIN</td>
<td>PUT INTEGER INTO INTEGER STRING</td>
</tr>
<tr>
<td>&amp;DAM.PUTNYB</td>
<td>PUT NON-NEG INTEGER INTO NYBLE OF BYTE STRING</td>
</tr>
<tr>
<td>&amp;DAM.PUTQBY</td>
<td>PUT INTEGER INTO QUADRUPLE BYTE OF BYTE STRING</td>
</tr>
<tr>
<td>&amp;DAM.SETQWO</td>
<td>SET QUARTER-WORD MODE</td>
</tr>
<tr>
<td>&amp;DAM.SLOBST</td>
<td>SLOPE (1ST DERIVATIVE) OF BYTE STRING</td>
</tr>
<tr>
<td>&amp;DAM.TRUAL</td>
<td>TRUE IF CST IS ALPHA (26 LTRS + SPACE)</td>
</tr>
<tr>
<td>&amp;DAM.TRUST</td>
<td>TRUTH VALUE OF CHARACTER STRING COMPARISON</td>
</tr>
<tr>
<td>Sort Routine Description</td>
<td>Source File(s)</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>Integer Bubble Sort Ascending</td>
<td>&amp;PRT.SC DAM.PREFACE-S</td>
</tr>
<tr>
<td>Integer Bubble Sort Descending</td>
<td>&amp;PRT.SC DAM.APPENDIX-S</td>
</tr>
<tr>
<td>Integer Hibbard’s Shell Sort Ascending</td>
<td>&amp;PRT.SC DAM.ISRTBA</td>
</tr>
<tr>
<td>Integer Hibbard’s Shell Sort Descending</td>
<td>&amp;PRT.SC DAM.ISRTBD</td>
</tr>
<tr>
<td>Integer Shuttle Sort Ascending</td>
<td>&amp;PRT.SC DAM.ISRTSA</td>
</tr>
<tr>
<td>Integer Shuttle Sort Descending</td>
<td>&amp;PRRT.S DAM.ISRSD</td>
</tr>
<tr>
<td>Taosort Using Hibbard’s Shell Sort</td>
<td>&amp;PRRT.S DAM.TSRTHS</td>
</tr>
</tbody>
</table>