THIS UFD CONTAINS ALL SOFTWARE UPDATES GENERATED AFTER
THE INITIAL REV. 16 RELEASE TO THE FIELD.
INFORMATION ABOUT ALL PREVIOUS UPDATE RELEASES SINCE
THE INITIAL RELEASE IS PRESENTED IN THIS FILE ALSO.
The initial Rev. 16 release was 16.3.
To update those files
Required on your master disk, FUTIL copy the program
Required to the UFD specified in the table under the -to- column
And use UPXXX as the program to copy and the name under
The name column as the name the program is to be copied as.

NOTE: ALL -TO- UFD'S MAY NOT EXIST ON YOUR
DISK IF YOU HAVE A 6 OR 12 MEG. BYTE DISK.

EXAMPLE: UPDATE NO. NAME TO
---------- ------ ----
UP001 CPUT1 T&M

FUTIL
> FROM 'THIS UFD' NOT NEEDED IF THIS IS HOME UFD
> TO T&M
> COPY UP001 CPUT1
> QU

NOTE: > EQUALS SUB-UFD IN -TO- COLUMN
NA EQUALS NOT ASSIGNED

USED ON (UFDNAME) DEFINITION
---------- ----------
8000 P8000 COBOL
8020 P8020 RJ2780
8060 P8060 RJCDC
8100 P8100 PRIMOS 4/5
8120 P8120 HASP300&400
8140 P8140 DBMS (DATABASE)
8150 P8150 RPG
8160 P8160 FORMS
8300 P8300 SPSS
8410 P8410 DPTX-DSC
8420 P8420 DPTX-TSF
8430 P8430 DPTX-TCF
8440 P8440 PRINET
8450 P8450 X.25
8520 P8520 BASICV

SET TABS 12 21 46 58 66 75
UPDATE NO. NAME TO SOURCE NO. SCN NO. DATE USED ON
## UFD Update Information File

**REV. 16.8F APRIL 20, 1979**

<table>
<thead>
<tr>
<th>File</th>
<th>Directory/Binary</th>
<th>Source</th>
<th>Date</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPAX-DSF</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>042079 8410</td>
<td></td>
</tr>
<tr>
<td>DPTX-DSF</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1328.000 0208</td>
<td>042079 8420</td>
<td></td>
</tr>
<tr>
<td>DPTX-DSF</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1324.002 0196</td>
<td>042079 8430</td>
<td></td>
</tr>
<tr>
<td>UP004-UP014 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP015 EDB</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP016 EDB</td>
<td>&lt;M164A1&gt;CMDC0</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP017-UP031 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP019 MAGSR</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP020 MAGRT</td>
<td>&lt;M164A1&gt;CMDC0</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP021 MAGSAV</td>
<td>&lt;M164A1&gt;CMDC0</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP022 MAGSAV</td>
<td>&lt;M164A1&gt;CMDC0</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP023-UP040 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP041 BASIC</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP042 BASIC</td>
<td>&lt;M164A1&gt;CMDC0</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP043 DBASIC</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP044 DBASIC</td>
<td>&lt;M164A1&gt;CMDC0</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP047 PRINET</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>042079 8440</td>
<td></td>
</tr>
<tr>
<td>UP048 X.25</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>042079 8450</td>
<td></td>
</tr>
<tr>
<td>UP049 FIXRAT</td>
<td>&lt;M164A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP050 FIXRAT</td>
<td>&lt;M164A1&gt;CMDC0</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP051 FIXRAT</td>
<td>&lt;M164A1&gt;CMDC0</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP052-UP063 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP064 ERDD.F</td>
<td>&lt;M164A1&gt;SYSCOM</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP065 ERDD.P</td>
<td>&lt;M164A1&gt;SYSCOM</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP066 SETSIZ</td>
<td>&lt;M164A1&gt;LIB7</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP067 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP068-UP070 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP071-UP072 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP073 PRMNT1</td>
<td>TMS400 (SOURCE)</td>
<td>SRC1326.003 0246</td>
<td>051079 8100</td>
<td></td>
</tr>
<tr>
<td>UP074 PRMNT1</td>
<td>T&amp;M (RUN)</td>
<td>SRC1326.003 0246</td>
<td>051079 8100</td>
<td></td>
</tr>
<tr>
<td>UP075 VTTYT1</td>
<td>TMS400 (SOURCE)</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP076 VTTYT1</td>
<td>T&amp;M (RUN)</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP077 C_VTTYT1</td>
<td>TMS400 (COMMAND FILE)</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP078-UP079 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP080 P4WCST</td>
<td>TMS400 (SOURCE)</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP082-UP085 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP086 CRTT1</td>
<td>T&amp;M (SOURCE)</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP087 CRTT1</td>
<td>T&amp;M (RUN)</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP090 DISCT1</td>
<td>TMSR1 (SOURCE)</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP091 DISUFD</td>
<td>T&amp;M (DIRECTORY)</td>
<td>SRC1326.003 0246</td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP094-UP096 SUPERCEDED</td>
<td></td>
<td></td>
<td>042079 8100</td>
<td></td>
</tr>
<tr>
<td>UP097 MIDAS</td>
<td>&lt;M165A1&gt;MFD</td>
<td>SRC1326.003 0246</td>
<td>254 072479 8100</td>
<td></td>
</tr>
<tr>
<td>UP098 KIDALB</td>
<td>&lt;M165A1&gt;LIB</td>
<td>SRC1326.003 0246</td>
<td>254 072479 8100</td>
<td></td>
</tr>
</tbody>
</table>

---

* REV. 16.5 JULY 24, 1979

---

**Not present on tape**
<p>| UP099 | KIDAFM  | <code>&lt;M165A1&gt;LIB (BINARY)</code> | 254 | 072479 | 8100 |
| UP100 | VKDALB  | <code>&lt;M165A1&gt;LIB (BINARY)</code> | 254 | 072479 | 8100 |
| UP101 | NKDALB  | <code>&lt;M164A1&gt;LIB (BINARY)</code> | 254 | 072479 | 8100 |
| UP102 | K4000   | <code>&lt;M165A1&gt;SYSTEM (BINARY)</code> | 254 | 072479 | 8100 |
| UP103 | K2014A  | <code>&lt;M165A1&gt;SYSTEM (BINARY)</code> | 254 | 072479 | 8100 |
| UP104 | K2014B  | <code>&lt;M165A1&gt;SYSTEM (BINARY)</code> | 254 | 072479 | 8100 |
| UP104A | IMIDAS  | <code>&lt;M165A1&gt;SYSTEM (BINARY)</code> | 254 | 072479 | 8100 |
| UP105 | CREATEK | <code>&lt;M165A1&gt;CMDNCO (RUN)</code> | 254 | 072479 | 8100 |
| UP106 | KBUILD  | <code>&lt;M165A1&gt;CMDNCO (RUN)</code> | 254 | 072479 | 8100 |
| UP107 | KIDDEL  | <code>&lt;M165A1&gt;CMDNCO (RUN)</code> | 254 | 072479 | 8100 |
| UP108 | REMAKE  | <code>&lt;M165A1&gt;CMDNCO (RUN)</code> | 254 | 072479 | 8100 |
| UP109 | MCLUP   | <code>&lt;M165A1&gt;CMDNCO (RUN)</code> | 254 | 072479 | 8100 |
| UP110 | C_MDLC1 | TMS400 (COMMAND) | 311 | 072479 | 8100 |
| UP111 | MDLCT1  | TMS400 (SOURCE) | SRC1316.003 | 311 | 072479 | 8100 |
| UP112 | MDLCT1  | T&amp;M (RUN) | 311 | 072479 | 8100 |
| UP113 | C_MDLC2 | TMS400 (COMMAND) | 259 | 072479 | 8100 |
| UP114 | MDLCT2  | TMS400 (SOURCE) | SRC1317.002 | 259 | 072479 | 8100 |
| UP115 | MDLCT2  | T&amp;M (RUN) | 259 | 072479 | 8100 |
| UP116 | C_MDLC3 | TMS400 (COMMAND) | 260 | 072479 | 8100 |
| UP117 | MDLCT3  | TMS400 (SOURCE) | SRC1318.002 | 260 | 072479 | 8100 |
| UP118 | MDLCT3  | T&amp;M (RUN) | 260 | 072479 | 8100 |
| UP119 | C_MDLC4 | TMS400 (COMMAND) | 261 | 072479 | 8100 |
| UP120 | MDLCT4  | TMS400 (SOURCE) | SRC1319.002 | 261 | 072479 | 8100 |
| UP121 | MDLCT4  | T&amp;M (RUN) | 261 | 072479 | 8100 |
| UP122 | C_MDLC5 | TMS400 (COMMAND) | 262 | 072479 | 8100 |
| UP123 | MDLCT5  | TMS400 (SOURCE) | SRC1320.003 | 262 | 072479 | 8100 |
| UP124 | MDLCT5  | T&amp;M (RUN) | 262 | 072479 | 8100 |
| UP125 | C_MDLC6 | TMS400 (COMMAND) | 314 | 072479 | 8100 |
| UP126 | MDLCT6  | TMS400 (SOURCE) | SRC1321.004 | 314 | 072479 | 8100 |
| UP127 | MDLCT6  | T&amp;M (RUN) | 314 | 072479 | 8100 |
| UP128 | C_MDLC7 | TMS400 (COMMAND) | 264 | 072479 | 8100 |
| UP129 | MDLCT7  | TMS400 (SOURCE) | SRC1322.002 | 264 | 072479 | 8100 |
| UP130 | MDLCT7  | T&amp;M (RUN) | 264 | 072479 | 8100 |
| UP131 | C_MDLC8 | TMS400 (COMMAND) | 315 | 072479 | 8100 |
| UP132 | MDLCT8  | TMS400 (SOURCE) | SRC1323.004 | 315 | 072479 | 8100 |
| UP133 | MDLCT8  | T&amp;M (RUN) | 315 | 072479 | 8100 |
| UP134 | C_MDLC9 | TMS400 (COMMAND) | 266 | 072479 | 8100 |
| UP135 | MDLCT9  | TMS400 (SOURCE) | SRC1327.001 | 266 | 072479 | 8100 |
| UP136 | MDLCT9  | T&amp;M (RUN) | 266 | 072479 | 8100 |
| UP137 | URC1    | T&amp;MRS1 (SOURCE) | SRC0732.006 | 248 | 072479 | 8100 |
| UP138 | URC1    | T&amp;M (RUN) | 248 | 072479 | 8100 |
| UP139-UP140 | SUPERCEDED | | | | |
| UP141 | HSSCT2  | T&amp;MRS (SOURCE) | SRC0796.007 | 252 | 072479 | 8100 |
| UP142 | HSSCT2  | T&amp;M (RUN) | 252 | 072479 | 8100 |
| UP143-UP146 | SUPERCEDED | | | | |
| UP146A | PXT1    | <code>&lt;M165A1&gt;TMS400 (SOURCE)</code> | SRC1304.007 | 255 | 072479 | 8100 |
| UP146B | PXT1    | <code>&lt;M165A1&gt;T&amp;M (RUN)</code> | 255 | 072479 | 8100 |
| UP146C | AMCT5   | <code>&lt;M165A1&gt;T&amp;MRS1 (SOURCE)</code> | SRC1325.002 | 253 | 072479 | 8100 |
| UP146D | AMCT5   | <code>&lt;M165A1&gt;T&amp;M (RUN)</code> | 253 | 072479 | 8100 |
| UP146E | FTL5700 | <code>&lt;M165A1&gt;TMS400 (DIRECTORY)</code> | SRC1335 | 256 | 072479 | 8100 |
| UP146F | FTL5700 | <code>&lt;M165A1&gt;T&amp;M (RUN)</code> | 256 | 072479 | 8100 |
| UP146G | P500T2  | <code>&lt;M165A1&gt;TMS400 (SOURCE)</code> | SRC1307.003 | 306 | 072479 | 8100 |
| UP146H | P500T2  | <code>&lt;M165A1&gt;T&amp;M (RUN)</code> | 306 | 072479 | 8100 |
| UP146I | CPUT4   | <code>&lt;M165A1&gt;TMS400 (SOURCE)</code> | SRC1334.001 | 309 | 072479 | 8100 |</p>
<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Date</th>
<th>Time</th>
<th>Size</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP146J</td>
<td>CPU4 &amp; M165A1-T&amp;M (RUN)</td>
<td></td>
<td>309</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP146K</td>
<td>XACHE1 &amp; M165A1-TMS400 (SOURCE)</td>
<td>SRC1305.005</td>
<td>310</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP146L</td>
<td>XACHE1 &amp; M165A1-T&amp;M (RUN)</td>
<td></td>
<td>310</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP147</td>
<td>FTN &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>267</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP148</td>
<td>FTN &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>267</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP149-150</td>
<td>SUPERCEDED</td>
<td></td>
<td></td>
<td>062679</td>
<td>8100</td>
</tr>
<tr>
<td>UP151-154</td>
<td>SUPERCEDED</td>
<td></td>
<td></td>
<td>062679</td>
<td>8100</td>
</tr>
<tr>
<td>UP155</td>
<td>BASICV &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>273</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP156</td>
<td>FUNCSX &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP157</td>
<td>IOCSV &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP158</td>
<td>FLIB1V &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP159</td>
<td>FLIB3V &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP160</td>
<td>IFTNLB &amp; M166A1-LIB (BINARY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP161</td>
<td>PFTNLB &amp; M166A1-LIB (BINARY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP162</td>
<td>NPFITNLB &amp; M166A1-LIB (BINARY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP163</td>
<td>S4000 &amp; M166A1-SYSTEM (BINARY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP164</td>
<td>S2014A &amp; M166A1-SYSTEM (BINARY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP165</td>
<td>S2014B &amp; M166A1-SYSTEM (BINARY)</td>
<td></td>
<td>274</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP166</td>
<td>FLIB6 &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>275</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP167</td>
<td>FTNL6 &amp; M166A1-LIB (BINARY)</td>
<td></td>
<td>275</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP168</td>
<td>FAM &amp; M166B1-PRINET (DIRECTORY)</td>
<td></td>
<td>276</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP169</td>
<td>EDB &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>277</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP170</td>
<td>*LIBEDB &amp; M166A1-LIB (RUN)</td>
<td></td>
<td>277</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP171</td>
<td>SEG &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP172</td>
<td>SEG &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP173</td>
<td>SHARE4 &amp; M166A1-LIB (BINARY)</td>
<td></td>
<td>278</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP174</td>
<td>LOAD &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>279</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP175</td>
<td>LOAD &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>279</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP176</td>
<td>DBMS &amp; M166B1-MFD (DIRECTORY)</td>
<td></td>
<td>280</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP177</td>
<td>FTOPT &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>281</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP178</td>
<td>FTOPT &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>281</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP179</td>
<td>SLIST &amp; M166A1-AIDS2 (SOURCE)</td>
<td></td>
<td>282</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP180</td>
<td>SLIST &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>282</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP181</td>
<td>MATHLB &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>283</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP182</td>
<td>MATHLB &amp; M166A1-LIB (BINARY)</td>
<td></td>
<td>283</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP183-185</td>
<td>SUPERCEDED</td>
<td></td>
<td></td>
<td>062679</td>
<td>8100</td>
</tr>
<tr>
<td>UP186</td>
<td>NEWSPL &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>285</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP187</td>
<td>SPOOL0 &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>285</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP188</td>
<td>SPOOL &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>285</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP189</td>
<td>SPOOLS &amp; M166A1-LIB (BINARY)</td>
<td></td>
<td>285</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP190</td>
<td>VSPOOS &amp; M166A1-LIB (BINARY)</td>
<td></td>
<td>285</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP191</td>
<td>MAGNET &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>286</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP192</td>
<td>MAGNET &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>286</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP193-196</td>
<td>SUPERCEDED</td>
<td></td>
<td></td>
<td>062679</td>
<td>8100</td>
</tr>
<tr>
<td>UP197</td>
<td>RUNOFF &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>288</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP198</td>
<td>RUNOFF &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>288</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP199</td>
<td>FUTIL &amp; M166A1-MFD (DIRECTORY)</td>
<td></td>
<td>289</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP200</td>
<td>FUTIL &amp; M166A1-CMDNCO (RUN)</td>
<td></td>
<td>289</td>
<td></td>
<td></td>
</tr>
<tr>
<td>File</td>
<td>Description</td>
<td>Revision</td>
<td>Date</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>----------</td>
<td>------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP201</td>
<td>FUTIL &lt;M166A1&gt;CMDNC0 (RUN)</td>
<td>289</td>
<td>062679 8100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP202</td>
<td>COBOL &lt;M166A1&gt;MFD (DIRECTORY)</td>
<td>290</td>
<td>062679 8000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP203</td>
<td>C4000 &lt;M166A1&gt;SYSTEM (BINARY)</td>
<td>290</td>
<td>062679 8000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP204</td>
<td>C2014A &lt;M166A1&gt;SYSTEM (BINARY)</td>
<td>290</td>
<td>062679 8000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP205</td>
<td>C2014B &lt;M166A1&gt;SYSTEM (BINARY)</td>
<td>290</td>
<td>062679 8000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP206</td>
<td>PXTI &lt;M166A1&gt;TMS400 (SOURCE) SRC1304.G07</td>
<td>255</td>
<td>062679 8100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP207</td>
<td>PXTI &lt;M166A1&gt;T&amp;M (RUN)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP208</td>
<td>AMLOCT5 &lt;M166A1&gt;T&amp;MRS1 (SOURCE) SRC1325.002</td>
<td>253</td>
<td>062679 8100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP209</td>
<td>AMLOCT5 &lt;M166A1&gt;T&amp;M (RUN)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP210</td>
<td>FLT750 &lt;M166A1&gt;TMS400 (DIRECTORY) SRC1335</td>
<td>256</td>
<td>062679 8100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP211</td>
<td>FLT750 &lt;M166A1&gt;T&amp;M (RUN)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UP212</td>
<td>C_FLT750 &lt;M166A1&gt;TMS400 (COMMAND)</td>
<td></td>
<td>062679 8100</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**REV. 16.8 SEPTEMBER 11, 1979**

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
<th>Revision</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP213</td>
<td>RJECOM &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>329</td>
<td>091179 RJECOM</td>
</tr>
<tr>
<td>UP214</td>
<td>RJ80 &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>330</td>
<td>091179 8020</td>
</tr>
<tr>
<td>UP215</td>
<td>RJ1004 &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>331</td>
<td>091179 8180</td>
</tr>
<tr>
<td>UP216</td>
<td>RJ200UT &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>332</td>
<td>091179 8060</td>
</tr>
<tr>
<td>UP217</td>
<td>RJ7020 &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>333</td>
<td>091179 RJ7020</td>
</tr>
<tr>
<td>UP218</td>
<td>RJGRTS &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>334</td>
<td>091179 RJGRTS</td>
</tr>
<tr>
<td>UP218A</td>
<td>RJHASP &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td></td>
<td>091179 8120</td>
</tr>
<tr>
<td>UP219</td>
<td>FIXRAT &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>335</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP220</td>
<td>COPY &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>335</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP220A</td>
<td>COPY &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>335</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP221</td>
<td>APPLIB &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>336</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP222</td>
<td>APPLIB &lt;M168A1&gt;LIB (BINARY)</td>
<td>336</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP223</td>
<td>APPLIB &lt;M168A1&gt;LIB (BINARY)</td>
<td>336</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP224</td>
<td>CX &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>337</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP225</td>
<td>CX** &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>337</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP226</td>
<td>CX &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>337</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP227</td>
<td>ED &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>342</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP228</td>
<td>ED &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>342</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP229</td>
<td>NSED &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>342</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP230</td>
<td>ED2000 &lt;M168A1&gt;SYSTEM (RUN)</td>
<td>342</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP231</td>
<td>FIXRAT &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>338</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP232</td>
<td>MAKE &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>338</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP233</td>
<td>MAKE &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>338</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP234</td>
<td>CMPF &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>339</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP235</td>
<td>MGF &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>339</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP236</td>
<td>PRI400 &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>340</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP237</td>
<td>PRI400 &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>340</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP238</td>
<td>PR20S &lt;M168A1&gt;MFD (DIRECTORY)</td>
<td>341</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP239</td>
<td>*DOS64 &lt;M168A1&gt;DOS (RUN)</td>
<td>341</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP240</td>
<td>*DOS64 &lt;M168B1&gt;DOS (RUN)</td>
<td>341</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP241</td>
<td>TDO64 &lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>341</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP242</td>
<td>MLCT5 &lt;M168A1&gt;TMS400 (SOURCE)</td>
<td>318</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP243</td>
<td>MLCT5 &lt;M168A1&gt;T&amp;M (RUN)</td>
<td>318</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP244</td>
<td>RTCT2 &lt;M168A1&gt;T&amp;MRC (SOURCE)</td>
<td>317</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP245</td>
<td>RTCT2 &lt;M168A1&gt;T&amp;M (RUN)</td>
<td>317</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP246</td>
<td>HSPT2 &lt;M168A1&gt;T&amp;MRS1 (SOURCE)</td>
<td>322</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP247</td>
<td>HSPT2 &lt;M168A1&gt;T&amp;M (RUN)</td>
<td>322</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP248</td>
<td>XACHE1 &lt;M168A1&gt;TMS400 (SOURCE)</td>
<td>323</td>
<td>091179 8100</td>
</tr>
<tr>
<td>UP249</td>
<td>XACHE1 &lt;M168A1&gt;T&amp;M (RUN)</td>
<td>323</td>
<td>091179 8100</td>
</tr>
</tbody>
</table>
**UFD UPDATE INFORMATION FILE REV. 16.8F**

<table>
<thead>
<tr>
<th>UP250</th>
<th>STLBT2</th>
<th>&lt;M168A1&gt;TMS400 (SOURCE)</th>
<th>324</th>
<th>091179</th>
<th>8100</th>
</tr>
</thead>
<tbody>
<tr>
<td>UP251</td>
<td>STLBT2</td>
<td>&lt;M168A1&gt;T&amp;M (RUN)</td>
<td>324</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP252</td>
<td>P500T2</td>
<td>&lt;M168A1&gt;TMS400 (SOURCE)</td>
<td>325</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP253</td>
<td>P500T2</td>
<td>&lt;M168A1&gt;T&amp;M (RUN)</td>
<td>325</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP254</td>
<td>CPUT4</td>
<td>&lt;M168A1&gt;TMS400 (SOURCE)</td>
<td>326</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP255</td>
<td>CPUT4</td>
<td>&lt;M168A1&gt;T&amp;M (RUN)</td>
<td>326</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP256</td>
<td>HSMT4</td>
<td>&lt;M168A1&gt;TMS400 (SOURCE)</td>
<td>327</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP257</td>
<td>HSMT4</td>
<td>&lt;M168A1&gt;T&amp;M (RUN)</td>
<td>327</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP257A</td>
<td>C_HSMT4</td>
<td>&lt;M168A1&gt;TMS400 (COMMAND)</td>
<td>321</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP258</td>
<td>XACHE2</td>
<td>&lt;M168A1&gt;TMS400 (SOURCE)</td>
<td>321</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP259</td>
<td>XACHE2</td>
<td>&lt;M168A1&gt;T&amp;M (RUN)</td>
<td>321</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP260</td>
<td>C_XACHE2</td>
<td>&lt;M168A1&gt;TMS400 (COMMAND)</td>
<td>321</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP261</td>
<td>HSMT3</td>
<td>&lt;M168A1&gt;TMSR1 (SOURCE)</td>
<td>328</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP262</td>
<td>HSMT3</td>
<td>&lt;M168A1&gt;T&amp;M (RUN)</td>
<td>328</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP263</td>
<td>PMA</td>
<td>&lt;M168A1&gt;MD (DIRECTORY)</td>
<td>343</td>
<td>091179</td>
<td>8100</td>
</tr>
<tr>
<td>UP264</td>
<td>PMA</td>
<td>&lt;M168A1&gt;CMDNC0 (RUN)</td>
<td>343</td>
<td>091179</td>
<td>8100</td>
</tr>
</tbody>
</table>

* END OF TABLE *

**REASON FOR CHANGE**

* TA 15 20 *

UP001 (DPTX-DSC) THIS IS A NEW PRODUCT FOR REV. 16.4.

UP002 (DPTX-TSF) THIS IS A NEW PRODUCT FOR REV. 16.4.

UP003 (DPTX-TCF) THIS IS A NEW PRODUCT FOR REV. 16.4.

UP004 (BASICV) TO FIX LEADING SPACES IN "WRITE USING" FORMAT STRING, WHICH WERE PREVIOUSLY IGNORED.

UP007 (APPLIB) 16.4 FIXES A MINOR BUG FOUND IN MSUBSA (MOVE SUBSTRING) WHICH CAUSED OVERLAPPING FIELDS TO BE MOVED INCORRECTLY.

UP008 (APPLIB) SEE UP007.

UP009 (VAPPLIB) SEE UP007.

UP010 (RUNOFF) (1) TARS 23221 & 23222 INVOLVING PROBLEMS WITH DECIMALIZATION COMMANDS PARTICULARLY "DL" (2) CAUSES "SM" COMMAND TO TAKE EFFECT ON NEXT PAGE RATHER THAN WRITING ON EXTRA PAGE. (3) NO LONGER SAVE A PLACE FOR THE PHANTOM HYPHENS IN THE TABLE OF CONTENTS. (4) STACK FILE NAMES FOR ERROR MESSAGES CORRECTLY.

UP011 (RUNOFF) SEE UP010.

UP012 (SEG) (1) ALLOW TREE NAME IN QUOTES AS INPUT TO CMDSEG.

(2) REINITIALIZE DEFAULT MODE IN 64V.

(3) FLAG 64R WHEN IN "NEVER 64" MODE.

(4) SHARE A SPLIT MODULE LESS THAN 4000 (BASE 8) WORDS IN LENGTH AND DISPLAY STACK ADDRESS DURING THE LOAD.
(5) HANDLE AN INTEGER COMMON BLOCK WITH THE DIMENSION 65536. NO TARS.

UP013 (SEG) SEE UP012.

UP014 (SHARE) SEE UP012.

UP015 (EDB) (1) FLAG SOURCE INPUT FILE AS A "BAD OBJECT FILE".
2) GENET (OBSOLETE BUT STILL SUPPORTED) NOW WORKS.

UP016 (EDB) SEE UP015.

UP017 (LOAD) (1) TAR 25536 DEFERRED COMMON ON A LIBRARY "COMMON" BLOCK BUG FIXED.
2) ALLOWS LARGER COMMON REDEFINITION WHEN DEFERRED.

UP018 (LOAD) SEE UP017.

UP019 (MGRS)

UP020 (MGRST) (1) HANDLES THE CONDITION THAT "A DATA RECORD FOLLOWS A UFD TREE NAME RECORD".
2) PRINT ERROR MESSAGE AND PAUSE WHEN A "DISC FULL" CONDITION OCCURS. (TAR 11969)
3) PRINT PATHNAME OF THE FILE AT THE TIME AN "UNEXPECTED EOF" CONDITION OCCURS.
4) SET READ/WRITE LOCK CORRECTLY. (TAR 10554)
5) REMOVE "-LONG" FROM USAGE LINE. (TAR 22800)

UP021 (MGRSA) (1) SAVE UFD WHICH HAS "READ ONLY" PERMISSION TO NON-OWNER AND FILES WITHIN THAT UFD WHICH PERMIT READ ACCESS TO NON-OWNER. PASSWORDS FOR THE SAVED UFD ARE SET TO NULL.
2) WHEN PROGRAM ASKS FOR A NEW TAPE, PROGRAM CHECKS TO SEE IF THE NEW TAPE IS AT LOAD POINT. IF NOT, AND THE TAPE IS THE SECOND PHYSICAL REEL OF A LOGICAL TAPE, PROGRAM WILL QUERY USER TO SEE IF HE WANTS THE TAPE TO BE Rewound. IF HIS ANSWER IS "YES", TAPE WILL BE Rewound. IF THE ANSWER IS "NO", PROGRAM WILL ASK FOR A NEW TAPE UNIT.

UP022 (MGRSA) SEE UP021.

UP023 (FTH) TAR 23673 GENERALIZED SUBSCRIPTS CAN GENERATE BAD CODE WHEN A VARIABLE IS SUBTRACTED FROM A CONSTANT.

TAR 25264 "LS" and "RS" INTRINSICS GENERATE BAD CODE FOR NEGATIVE SHIFT COUNTS.

TAR 25564 THE COMPILER HANGS WHEN IN 64V MODE A STATEMENT FUNCTION IS PASSED AS AN OCTAL ARGUMENT.

WHEN A "INSERT" FILE IS NOT FOUND, THE ERROR MESSAGE WILL NOT CONTAIN A SPURIOUS "T". THE "SHORTCALL" STATEMENT WORKS WITH LIBRARY CONVERSION FUNCTIONS.

MINOR PROBLEMS IN PARSING ARRAY REFERENCES AND
STATEMENT FUNCTIONS ARE FIXED. THE COMPILER USED TO GET THE EXCESS SUBSCRIPTS AND TOO FEW SUBSCRIPTS ERROR MESSAGES REVERSED.

(FTN) SEE INFO ON UP023.

(FINOPT) ALL THE FIXES FOR "FTN" APPLY TO "FTNOPT" AS WELL. OPTIMIZER PROBLEMS WHICH HAVE BEEN FIXED ARE:

- USE OF THE DO LOOP OPTIMIZER SOMETIMES PRODUCED LESS EFFICIENT CODE OUTSIDE LOOPS.
- TEMPORARY VARIABLES INSIDE OPTIMIZED DO LOOPS WERE NOT ALWAYS FREED PROPERLY.
- OPTIMIZED DO LOOPS OCCASIONALLY HAD BAD CODE FOR MIXED MODE ARITHMETIC.

(FINOPT) SEE INFO ON UP025.

(CUBOL) TO CORRECT TAR 25666. QUALIFIED DATA NAMES NOT OPERATING CORRECTLY.

(C4000) SEE UP027.

(C2014A) SEE UP027.

(C2014B) SEE UP027.

(FLIBGV) [F$IO] FREE FORMAT COMPLEX INPUT DID NOT WORK FOR F$IO.

(VDSPKS) ETSRC$$3 ">A" DID NOT WORK FOR TSRC$$.

(DDSPKS) SEMLIB P300 CODE REMOVED. (TAR 81470).

(TSRC$$ ">A" DID NOT WORK.

(P300 CODE REMOVED (TAR 81470)

(">A" DID NOT WORK.

(PFTNLIB) SEE UP034.

(NPFTNLIB) SEE UP034.

(FTNLIB) SEE UP034.

(S4000) SEE UP034.

(S2014A) SEE UP034.

(S2014B) SEE UP034.

(BASIC) TARS 12546 & 80852 "PRINT USING" JUXTAPOSED.
ITEMS WHEN THE FIRST NUMERIC ITEMS OVERFLOWED.
TAR 13717 ".NL." DID NOT RESET THE COLUMN COUNT
IN ENTER STATEMENT.
TAR 24728 STATEMENT NUMBER "0" WAS NOT SENSED AS AN ERROR.
TAR 15819 "PRINT USING" ROUNDEING IS NOT CONSISTENT.
MACHINE FLOATING ACCURACY IS THE PROBLEM HERE, BUT NOTE THAT THE ACTUAL COMPUTATION ACCURACY IS NOT AFFECTED BY THIS PROBLEM, WHICH IS DUE TO THE INPUT CONVERSION IF ASCII DIGITS TO FLOATING NUMBERS. A BETTER METHOD IS USED BY BASIC/VM AND FORTRAN, SO THESE PROBLEMS WILL NOT SHOW UP.
TAR'S 80236 & 80469 "HALT"'S ARE ENCOUNTERED WHEN STRINGS ARE PASSED TO A FORTRAN PROGRAM. THE DOCUMENTATION IS WRONG AND INDEED STRINGS ARE NOT ALLOWED TO BE PASSED TO A FORTRAN PROGRAM.
TAR 22783 A "FOR-NEXT" UNMATCHING ERROR WAS GENERATED WHEN IN FACT NO MISMATCH EXISTED.

UP042 (BASIC) SEE INFO ON UP041.
* UP043 (DBASIC) SEE INFO ON UP041.
* UP044 (DBASIC) SEE INFO ON UP041.
* UP045 (PRI400)

BUG FIXES AT REV. 16.4

COMINPUT COMMAND

THE FILE UNIT SPECIFIED WAS IGNORED IF SPECIFIED AFTER A "OPTION. E.G., IF THE COMMAND 'CO - CONTINUE 7' WAS GIVEN, FILE UNIT 6 WAS USED. (TAR 80697)

FILUNIT COLD START PARAMETER

IF A FILUNIT PARAMETER WAS USED IN THE COLD START FILE, SPURIOUS RESULTS WOULD OCCUR.

ASSIGNED AMLC LINES

OUTPUT CHARACTERS COULD BE LOST WHEN UNASSIGNING AMLC LINES. (TAR 23415)

WTILINES

DATE-TIME MODIFIED NOT UPDATED WHEN FILE ACCESSED WITH CALL TO WTILINES.

SHARE
IT WAS NOT POSSIBLE TO SHARE AN ENTIRE SEGMENT, I.E.,
RESTORE FILE WHOSE START ADDR = 0 AND END ADDR = 177777
OCTAL. (TAR 10555)

DID NOT GIVE ERROR MESSAGE IF FILE SPECIFIED WAS A
DIRECTORY. COMMAND OF FORM "COMO TREENAME -O" WOULD NOT
WORK.

-DUE TO A CONFLICT WITH PREVIOUSLY DEFINED HARDWARE
DEVICE ADDRESSES, THE DEVICE ADDRESS OF THE PRIMENET
NODE CONTROLLER (PNC) HAS BEEN CHANGED FROM '61 TO '07.

UP046
(PRIRUN) SEE UP045.

*  

UP047
(PRINET) FAM FOR REV. 16.4. THE FOLLOWING BUGS HAVE BEEN
FIXED:
-ACCESSING SEGMENT DIRECTORIES VIA PATHNAME NOW WORKS.
(I.E., SEE REMOTE_UFD>SUBUFD>PROG )
-DUPLICATE RECEIVED MESSAGE BUG IS PROBABLY FIXED.
-LONG WRITE LINES NOW WORK WITH > 255 TRAILING SPACES.
-GROSS FLAG IS NOW RESET IN FAMCYL, (COULD GET LOCKED
SET IN 16.2).
-FAM NOW ACCEPTS COS CODES TO WORK WITH PRIMENET CIRCUIT
CLEARING CAUSES.
-THE INTERNAL VERSION NUMBER AND RECEIVE BLOCK SIZE PASSING
HAS BEEN UPDATED TO CONFORM WITH 17.0'S EXPECTATIONS.

UP048
(X.25) NETCNF HAS BEEN FIXED FOR HETEROGENEOUS COMBINATIONS
OF PRIMENET AND X.25 SOFTWARE IN THE SAME NETWORK. IT
IS NO LONGER A REQUIREMENT THAT IF ANY NODE HAS THE X.25
SOFTWARE, THEY ALL MUST HAVE IT. TO SUPPORT THIS FEATURE
THERE HAVE BEEN SOME INTERNAL CHANGES TO THE FORMAT OF
THE CONFIGURATION FILE 'NETCON'.

UP049
(FIXRAT) UFD COMPRESSION FAILED TO WORK CORRECTLY.

UP050
(FIXRAT) SEE UP049.

*  

UP051
(FIXRAT) SEE UP049.

UP052
(MIDAS)

ABSTRACT

NEW AT REV 16.4, MIDAS UTILITY *MPACK SORTS DATA RECORDS BY PRIMARY KEY
AND RECOVERS SPACE OCCUPIED BY DATA RECORDS WHICH HAVE BEEN MARKED FOR
DELETION.

FOR REV 16 MIDAS FILES, *MPACK SORTS DATA RECORDS BY PRIMARY KEY AND
RECOVERS SPACE OCCUPIED BY DATA RECORDS WHICH HAVE BEEN MARKED FOR
DELETION. INDEXES ARE ALSO RESTRUCTURED SO THAT THEY OCCUPY AS LITTLE
DISK SPACE AS POSSIBLE. *MPACK IS USEFUL FOR APPLICATIONS IN WHICH 1) DISK SPACE IS VERY LIMITED, AND/OR 2) RECORDS ARE OFTEN INSERTED AND DELETED FROM A MIDAS FILE.

*MPACK IS BUILT IN UFD MIDAS>SOURCE. NOTE THAT *MPACK IS BUILT IN UFD MIDAS>LD, NOT CMDNCO, AND EXECUTES IN R-MODE ONLY. *MPACK HAS BASICALLY TWO OPTIONS. A MIDAS FILE MAY SIMPLY BE RESTRUCTURED. IN THIS CASE THE EXISTING FILE IS OVERWRITTEN WITH THE RESTRUCTURED DATA. THE SECOND OPTION CAUSES THE RESTRUCTURED DATA TO BE WRITTEN TO A SECOND FILE, THUS PRESERVING THE ORIGNAL FILE. Figure 1 ILLUSTRATES HOW TO USE *MPACK. COMMENTS ARE ENCLOSED IN PARENTHESSES AND USER INPUT IS UNDERLINED.

OK, R *MPACK
GO
*MPACK REV 16.4] ENTER MIDAS FILE NAME: ACCT>MASTER (PATH NAME OF FILE TO BE ) (RESTRUCTURED)
 + OK TO OVERWRITE THE FILE? NO (SEE NOTE 1. )
 + ENTER NEW MIDAS FILE NAME: FILE1 (PATH NAME OF FILE TO CONTAIN THE) (RESTRUCTURED INFORMATION.)
 + FILE ALREADY EXISTS. OK TO OVERWRITE? NO (SEE NOTE 2. )
 + ENTER NEW MIDAS FILE NAME: FILE2 (SEE NOTE 3. )
 +
 BEGIN PROCESSING INDEX 0 AT 11:22:00
 ENTRIES INDEXED: 250
 BEGIN PROCESSING INDEX 1 AT 11:26:27
 ENTRIES INDEXED: 92
 RESTRUCTURE COMPLETED AT 11:28:26

Figure 1

NOTES

1. THE NO RESPONSE INDICATES THAT THE RESTRUCTURED DATA SHOULD BE WRITTEN TO ANOTHER FILE. THE FILE, MASTER, WAS NOT MODIFIED.

2. THE NO RESPONSE INDICATES THAT THE MIDAS FILE, FILE1, SHOULD NOT BE USED. *MPACK ALSO VERIFIES THAT THE FILE IS A VALID MIDAS FILE. IF NOT VALID, *MPACK NOTIFIES THE USER AND REQUESTS A NEW PATH NAME.

3. SINCE FILE2 DID NOT EXIST, *MPACK CREATED IT.

*UP053 (KIDALD) SEE UP052.
UP054 (K1DAFM) SEE UP052.
UP055 (VKDALB) SEE UP052.
UP056 (NVKDALB) SEE UP052.
UP057 (K4000) SEE UP052.
UP058 (K2014A) SEE UP052.
UP059 (K2014B) SEE UP052.
UP060 (CREATK) SEE UP052.
UP061 (KBUILD) SEE UP052.
UP062 (KIDDEL) SEE UP052.
UP063 (REMAKE) SEE UP052.
UP064 (ERRDP) ERROR CODE FOR DPTX.
UP065 (ERRDP) SEE INFO ON UP064.
UP066 (SETSIZ) SETSIZ SOMETIMES WENT INTO AN INFINITE LOOP UNDER PRIMOS 2.

UP067 (DBMS) THE FOLLOWING IS A LIST OF BUGS FIXED IN REV. 16.3.
EXCEPT WHERE NOTED, THE BUGS WERE FIXED BASED ON INTERNAL ERRORS OR ERRORS THAT WERE REPORTED BY CMSI OVER THE PHONE AND THERE ARE NO TAR NUMBERS.

1) THE FOLLOWING PATCHES HAVE BEEN MADE TO DMLCP.
   A. THE SIZE OF THE INTERNAL RECORD AREA HAS BEEN EXPANDED FROM 8KB TO 32 KB TAR 24722.
   B. THE OPEN COMMAND WILL NOW ONLY OPEN AREAS SPECIFIED ON THE OPEN COMMAND RATHER THAN ALL AREAS.
   C. THE CLEAR_ERROR_COMMAND HAS BEEN FIXED SO THE SYSTEM WILL NOT HANG.
   D. THE 710F ERROR IN THE ROUTINE SETSLST HAS BEEN FIXED.
   E. THE ROUTINE PUTSLST HAS BEEN PATCHED SO THAT DUPLICATES WILL BE INSERTED IN THE PROPER ORDER.
   F. AFTER IMAGE LOGGING HAS BEEN PATCHED TO ACCOMODATE BUCKETS LARGER THAN ONE (1) PAGE.
   G. R4VAL HAS BEEN PATCHED TO ACCOMODATE LONG RETRIEVAL TRANSACTIONS.

2) CLUP HAS BEEN PATCHED SO THAT CERTAIN ERRORS WILL BE DISPLAYED ON THEIR TERMINAL WHEN THEY OCCUR.

3) DBACP HAS BEEN FIXED SO THAT IT MAY INITIALIZE A FILE LARGER THAN 32,000 BLOCK PROPERLY.
UP068  (CPUT4) TO REDUCE THE NUMBER OF TEST PROGRAMS. P4Q0T2 & P500T4 ARE COMBINED IN AND ARE REPLACED BY THIS NEW TEST.

UP069  (C-CPUT4) SEE UP068.

UP070  (CPUT4) SEE UP068.

UP071  (RTCT2) TO ENABLE THE TEST TO RUN ON A VCP AS WELL AS A SGC.

UP072  (RTCT2) SEE UP071.

UP073  (PRMNT1) ADDED TESTS IN ORDER TO TEST PARTS OF THE HARDWARE THAT WEREN'T PREVIOUSLY TESTED. TO HAVE COMPATIBILITY BETWEEN THE WIRE WRAP AND ETCH VERSIONS SO THAT THEY CAN RUN ON THE SAME PROGRAM.

DEVICE ADDRESS OF PRIMENET NODE CONTROLLER IS BEING CHANGED FROM '61 TO '07.

A BUG WAS FOUND WHEN TRYING TO LOAD THE A REGISTER WITH THE DEVICE ADDRESS PRIOR TO RUNNING THE PROGRAM.

UP074  (PRMNT1) SEE UP073.

UP075  (VTYT1) THIS DIAGNOSTIC CHECKS OUT THE SERIAL INTERFACE CAPABILITIES OF THE VCP V.I.A. PFO. THIS TEST OPERATED SIMILARLY TO TTYT2.

UP076  (VTYT1) SEE UP075.

UP077  (URCT1) SUPPORT OF VRC / DECISION DATA CARD PROCESSOR.

UP078  (URCT1) SEE UP077.

UP079  (P4WCST) TEST FAILED IF THERE WERE LESS THAN 64K OF MEMORY.

UP080  (P4WCST) SEE UP079.

UP081  (STLBT2) TO ACCOMMODATE THE P750.

UP082  (STLBT2) SEE UP081.

UP083  (PXT1) TO FIX STRING PROBLEM.

UP084  (PXT1) SEE UP083.

UP085  (CRTT1) (1) TO ADD A ROUTINE TO CHECK THE ABILITY FOR THE DEVICE TO TRANSMIT ON REQUEST BY THE HOST CPU AND CHECK THE INTEGRITY OF THE TERMINAL'S OWN MEMORY.

(2) TO CONDENSE THE WHOLE TEST INTO A SMALLER PACKAGE WHILE IMPROVING THE EFFECTIVENESS OF THE WHOLE TEST.

(3) TO REMOVE POSSIBLE BUG WHERE AMIC IS SHUTDOWN BEFORE IT HAS TIME TO CLEAR DEDICATED PELL.

UP086  (CRTT1) SEE UP085.
UP087 (AMLCT5) TO INCORPORATE TIMING CHANGES CAUSED BY THE VCP.

UP088 (AMLCT5) SEE UP087.

UP089 (DISCT1) TO INCORPORATE TIMING CHANGES CAUSED BY THE VCP.

UP092-UP096 (SPOOL) BETTER "QUEUE FULL" ERROR MESSAGE. (TAR 22414)

(2) HASP CONTROL ON SERIAL PRINTER. (TAR 23467)
CONCURRENT PROCESS HANDLING AND THE DETECTION AND CORRECTION OF CONCURRENCY ERRORS ARE THE TWO MAJOR AREAS OF MODIFICATION IN MIDAS AT REV 16.5. DESIGNED TO PROVIDE A SUBSTANTIAL PERFORMANCE IMPROVEMENT, THE NEW CONCURRENT PROCESS HANDLING METHOD WILL REQUIRE MODIFICATION OF FORTRAN AND PMA MIDAS APPLICATION PROGRAMS. THE NEW METHOD IS AVAILABLE TO COBOL USERS AT THIS RELEASE, TO BASIC USERS AT REV 16.6, AND TO RPG II USERS AT REV 17.1. USERS MAY EASILY DISABLE THE NEW METHOD AND, AS A RESULT, EMPLOY THE CONCURRENT PROCESS HANDLING METHOD AVAILABLE IN PREVIOUS RELEASES. NOTE THAT USERS WITH APPLICATIONS WHICH ACCESS MIDAS FILES OVER PRIME NET MUST DISABLE THE NEW CONCURRENT PROCESS HANDLING METHOD.

THE SECOND CHANGE, INDEPENDENT OF THE FIRST, ALLOWS MIDAS IN MOST CASES TO DETECT AND CORRECT CONCURRENCY ERRORS.

SECTION 2 DISCUSSES THE NEW CONCURRENT PROCESS HANDLING METHOD AND ITS IMPACT ON USER APPLICATIONS AND OPERATIONS. SECTION 3 DESCRIBES HOW MIDAS DETECTS AND CORRECTS CONCURRENCY ERRORS. INSTALLATION METHODS AND CONSIDERATIONS ARE DISCUSSED IN SECTION 4.
THIS PAGE RESERVED FOR THE TABLE OF CONTENTS.
1 INTRODUCTION

MIDAS AT REV 16.5 OFFERS FORTRAN AND PMA USERS TWO INDEPENDENT IMPROVEMENTS. FIRST, MANY USER APPLICATIONS MAY BE ABLE TO OPERATE SUBSTANTIALLY FASTER. TABLES 1.1 AND 1.2 SHOW SOME SAMPLE DATA. THE TEST PROGRAM PROCESSED A SINGLE MIDAS FILE CONTAINING 500 RECORDS. EACH RECORD WAS THE CONCATENATION OF FOUR ASCII TEN CHARACTER KEYS. FOR EACH RECORD, THE PROGRAM:

1) READ NEXT RECORD (OR FIRST) VIA PRIMARY KEY,
2) FOR EACH SECONDARY INDEX:
   2A) READ THE RECORD VIA THE SECONDARY KEY,
   2B) DELETED THE CURRENT KEY VALUE,
   2C) RE-INSERTED THE KEY VALUE.

THE PERFORMANCE DATA WERE OBTAINED ON A P-650 WITH 1024K BYTES OF MEMORY. MIDAS PROCESSES EXECUTED WITH THE FAM AND SPOOL PROCESSES AND A TERMINAL PROCESS. DATA IN TABLE 1.1 WERE OBTAINED FROM PROCESSES OPERATING CONCURRENTLY ON THE SAME MIDAS FILE. TABLE 1.2 SHOWS RESPONSE TIMES FOR CONCURRENT PROCESSES EXECUTING THE SAME TEST PROGRAM BUT OPERATING ON DIFFERENT COPIES OF THE SAME DATA.

<table>
<thead>
<tr>
<th>NUMBER OF CONCURRENT PROCESSES</th>
<th>REV 16.4</th>
<th>REV 16.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.7</td>
<td>0.4</td>
</tr>
<tr>
<td>2</td>
<td>2.2</td>
<td>0.8</td>
</tr>
<tr>
<td>3</td>
<td>3.7</td>
<td>1.2</td>
</tr>
<tr>
<td>4</td>
<td>5.1</td>
<td>1.6</td>
</tr>
<tr>
<td>5</td>
<td>6.9</td>
<td>2.0</td>
</tr>
<tr>
<td>6</td>
<td>---</td>
<td>2.5</td>
</tr>
<tr>
<td>7</td>
<td>---</td>
<td>3.0</td>
</tr>
</tbody>
</table>

TABLE 1.1 -- AVERAGE RESPONSE TIME PER RECORD PROCESSED (SECONDS) PROCESSES OPERATING ON THE SAME MIDAS FILE
TABLE 1.2 -- AVERAGE RESPONSE TIME PER RECORD PROCESSED (SECONDS) PROCESSES OPERATING ON DIFFERENT FILES.

DATA FOR COLUMN TWO OF TABLE 1.1 AND COLUMN THREE OF TABLE 1.2 WAS OBTAINED BY MODIFYING THE TEST PROGRAM TO CALL THE NEW MIDAS USER INTERFACE ROUTINES, OPENMS AND CLOM5$ RATHER THAN SRCH$. 

TO OBTAIN THIS PERFORMANCE INCREASE, MIDAS NOW USES A DIFFERENT METHOD OF HANDLING CONCURRENT PROCESSES. THIS NEW METHOD, HOWEVER, WILL REQUIRE CHANGES IN FORTRAN AND PMA AND APPLICATION PROGRAMS IN ORDER FOR THE PROGRAMS TO OBTAIN THE PERFORMANCE INCREASE. COBOL PROGRAMS, HOWEVER, REQUIRE NO CHANGES. USER OPTIONS ARE DETAILED IN SECTION 2.3.1. NOTE THAT UNMODIFIED PROGRAMS WILL STILL OPERATE AND THAT PROGRAMS NEED NOT ALL BE MODIFIED AT THE SAME TIME. HOWEVER, ALL FORTRAN AND PMA PROGRAMS WHICH USE THE UNSHARED MIDAS LIBRARIES (KIDALB AND NVDALD$) MUST BE RELOADED WHETHER OR NOT THE PROGRAMS ARE MODIFIED. COBOL PROGRAMS WHICH USE THE UNSHARED COBOL AND/OR MIDAS LIBRARIES MUST ALSO BE RELOADED.

THE SECOND IMPROVEMENT IN MIDAS IS COMPLETELY INDEPENDENT OF THE FIRST AND REQUIRES NO CHANGES IN APPLICATION PROGRAMS. MIDAS WILL NOW DETECT AND CORRECT CONCURRENCY ERRORS. THESE ERRORS MAY OCCUR WHEN THE POSITION OF A PROCESS IN A MIDAS FILE IS MODIFIED BY THE ACTION OF A CONCURRENT PROCESS. THE ONLY CASE THAT APPLICATION PROGRAMS MUST BE ABLE TO HANDLE OCCURS WHEN A PROCESS ATTEMPTS TO OPERATE ON ITS 'CURRENT RECORD' (EG. UPDATE IT) AND A CONCURRENT PROCESS HAS DELETED THE RECORD. IN THIS SPECIAL CASE MIDAS WILL DETECT THE 'ERROR' AND RETURN A STATUS CODE OF 13, WHICH NOW HAS A DIFFERENT MEANING FOR ERROR RECOVERY THAN STATUS CODE 13 AT REV 16.4.
2 HANDLING OF CONCURRENT MIDAS PROCESSES

2.1 OVERVIEW

In order to provide increased performance, MIDAS now employs a method of handling concurrent processes which differs from previous releases. In the past MIDAS coordinated concurrent processes by gating processes at the segment subfile level (e.g., a MIDAS file index). This method relied upon file system read/write locks and required that segment subfiles be opened at the start of each MIDAS file operation and closed upon completion of the operation. For example, to retrieve a record, MIDAS opened the index segment subfile(s) and the data segment subfile. When the retrieval completed, MIDAS closed these segment subfiles.

The new concurrent process handling method provides improved performance by greatly reducing the number of file system calls. Through use of a semaphore and a "lock" in shared memory, MIDAS simply allows only one process at a time to execute a MIDAS file operation. Therefore, MIDAS segment subfiles need not be closed at the end of each operation only to be reopened at the start of the next call. Details of the new method are described in section 2.2.

The new method of handling concurrent processes requires that MIDAS be notified both when a process is to begin using a MIDAS file and when the process has completed operations on the file. For FORTRAN and PMA users of the MIDAS call level interface, this requirement means that application programs must be modified. Section 2.3 describes methods of making these changes. Important installation instructions are detailed in section 4. It should be noted that PRIMENET users and users who do not wish to make application program changes may disable the new method of handling concurrent processes and thus return to the method employed by previous MIDAS releases. The procedure for disabling the new method is described in section 4.3.

2.2 IMPLEMENTATION METHOD

To maintain file integrity, MIDAS must synchronize concurrent processes. In previous releases of MIDAS, this synchronization was accomplished by opening file segments for reading and writing. Since file read/write locks were set to 2 (n readers and one writer), only one process could access a file segment at a time. A second process was only able to proceed when the first process finished its MIDAS operation and the file segments were closed. This method of synchronization required many calls to the file system routine SRCH$$ to open and close file segments and thus imposed a significant performance penalty.

In this release MIDAS does not close file segments between MIDAS operations. This, however, requires that MIDAS file read/write locks be set to 3 (n readers and m writers). Otherwise, concurrent processes would be unable to open a file segment which had been
ALREADY OPENED BY ANOTHER PROCESS. NOTE THAT IN ALL PAST AND
PRESENT RELEASES, MIDAS MAY WRITE INTO A FILE ON BEHALF OF A
USER-LEVEL READ REQUEST.

WITH FILE READ/WRITE LOCKS SET TO 3, FILE INTEGRITY COULD BE
DESTROYED. THIS WOULD HAPPEN, FOR INSTANCE, IF TWO PROCESSES BOTH
READ THE SAME RECORD AND THEN BOTH UPDATE THE RECORD. IN THIS CASE
THE FIRST UPDATE WOULD BE LOST. TO PREVENT LOSS OF FILE INTEGRITY,
MIDAS EMPLOYS A METHOD OF HANDLING CONCURRENT PROCESSES WHICH DOES
NOT DEPEND ON OPENING AND CLOSING FILE UNITS.

IN THE NEW METHOD WHEN MIDAS IS CALLED, A CHECK IS DONE TO SEE IF
ANY OTHER PROCESS IS USING MIDAS. TO DO THIS CHECK, MIDAS TESTS A
"LOCK" LOCATED IN A SHARED MEMORY SEGMENT. A ZERO VALUE INDICATES
THAT MIDAS IS AVAILABLE. IF NON-ZERO, THE LOW ORDER 15 BITS IS THE
USER NUMBER OF THE PROCESS CURRENTLY ACCESSING MIDAS. (NOTE: BIT
ONE IS ALWAYS SET WHEN MIDAS IS IN USE.) WHEN THE RESULT OF THE
LOCK TEST IS ZERO, THE LOCK IS SET TO INDICATE THAT THE CURRENT
PROCESS (DOING THE CHECK) NOW HAS SOLE ACCESS TO MIDAS. THIS "TEST
AND SET" OPERATION IS NON-INTERRUPTIBLE. THEREFORE A PROCESS CANNOT
MODIFY THE LOCK VALUE BETWEEN THE TIME THAT ANOTHER PROCESS HAS
TESTED AND SET THE LOCK VALUE. IF THE TEST AND SET OPERATION IS
SUCCESSFUL, THE PROCESS IS SAID TO HAVE "OBTAINED" THE LOCK.

IF WHEN TESTED, THE LOCK IS NON-ZERO, THE TESTING PROCESS MUST WAIT
UNTIL MIDAS BECOMES AVAILABLE. TO ACCOMPLISH THIS, THE PROCESS IS
SUSPENDED AND PUT ON A SEMAPHORE WAIT LIST. THE WAIT LIST FORMS A
QUEUE OF PROCESSES WAITING TO BEGIN A MIDAS OPERATION. EACH TIME AN
OPERATION COMPLETES, THE LOCK IS RELEASED, IE. THE LOCK VALUE IS
SET TO ZERO. A PROCESS IS THEN REMOVED FROM THE WAIT LIST. THE
RESTARTED PROCESS AGAIN MUST ATTEMPT TO OBTAIN THE LOCK.

2.3 APPLICATION IMPLICATIONS

2.3.1 USER OPTIONS

A USER HAS TWO BASIC OPTIONS WITH THE NEW MIDAS RELEASE.

1) THE USER MAY DISABLE THE NEW METHOD OF CONCURRENT
PROCESS HANDLING AND MAKE NO APPLICATION PROGRAM
CHANGES. ALTHOUGH THERE WOULD BE NO PERFORMANCE
GAIN, THE DETECTION AND CORRECTION OF CONCURRENCY
ERRORS WOULD STILL OCCUR. NOTE THAT THIS IS THE
ONLY OPTION AVAILABLE TO PRIMENET USERS.

2) THE USER MAY MODIFY SOME OR ALL APPLICATION PROGRAMS
IN ORDER TO SELECTIVELY OBTAIN A PERFORMANCE
IMPROVEMENT. UNMODIFIED PROGRAMS AUTOMATICALLY
USE THE NEW METHOD OF HANDLING CONCURRENT
PROCESSES BUT MAY SUFFER SOME PERFORMANCE
DEGRADATION.
2.3.2 APPLICATION PROGRAM MODIFICATIONS

WHEN MIDAS IS INSTALLED, USERS MUST RELOAD ALL APPLICATION PROGRAMES WHICH USE AN UNSHARED MIDAS LIBRARY. IN ADDITION, TO OBTAIN THE POTENTIAL PERFORMANCE INCREASE, USERS MUST MODIFY FORTRAN AND PMA MIDAS APPLICATION PROGRAMES. THE MODIFICATIONS INVOLVE INSERTING SUBROUTINE CALLS TO NOTIFY MIDAS THAT FILE SEGMENTS ARE NOT TO BE CLOSED BETWEEN CALLS TO MIDAS. NOTE THAT NOT ALL APPLICATIONS NEED BE MODIFIED AT THE SAME TIME.

USERS MAY CHOOSE FROM TWO METHODS OF PROGRAM MODIFICATION. THE FIRST METHOD INVOLVES INSERTING CALLS TO SUBROUTINE NTFYMS. THE FIRST CALL SHOULD BE INSERTED FOLLOWING THE CALL TO OPEN THE MIDAS FILE BUT BEFORE THE FIRST MIDAS FILE OPERATION. THE OTHER CALL TO NTFYMS SHOULD BE INSERTED JUST BEFORE THE CALL TO CLOSE THE MIDAS FILE. NTFYMS NOTIFIES MIDAS THAT A MIDAS FILE HAS JUST BEEN OPENED OR IS ABOUT TO BE CLOSED. FOR FURTHER DETAILS REFER TO THE SECTION WHICH DESCRIBES SUBROUTINE NTFYMS.

THE SECOND METHOD IS TO REPLACE THE CALLS WHICH OPEN AND CLOSE A MIDAS FILE WITH CALLS TO OPENMS$ AND CLOSMS$ RESPECTIVELY. SUBROUTINE OPENMS$ OPENS A MIDAS FILE AND THEN CALLS NTFYMS$. CLOSMS$ CALLS SUBROUTINE NTFYMS$ AND THEN CLOSES A MIDAS FILE. DETAILS ARE PROVIDED IN THE SECTIONS WHICH DESCRIBE OPENMS$ AND CLOSMS$.

MIDAS SUPPORTS R MODE APPLICATIONS. HOWEVER, BECAUSE THE R MODE MIDAS LIBRARY ENTERS V MODE TO DO A PORTION OF THE CONCURRENT PROCESS HANDLING, MIDAS WILL NOT WORK ON A PRIME P-300.
FUNCTION

NOTIFY MIDAS THAT A MIDAS FILE (SEGMENT DIRECTORY) HAS BEEN OPENED OR IS ABOUT TO BE CLOSED BY THE USER.

CALLING SEQUENCE

CALL NTFYMS (KEY, UNIT, STATUS)

KEY -- (INPUT) SPECIFIES WHETHER THE FILE HAS BEEN OPENED OR IS ABOUT TO BE CLOSED.
1 - FILE HAS BEEN OPENED
2 - FILE IS ABOUT TO BE CLOSED

UNIT -- (INPUT) FILE UNIT ON WHICH THE FILE IS OPEN

STATUS -- (OUTPUT) ERROR STATUS
0 - NO ERROR
10001 - BAD PARAMETER
10002 - TOO MANY MIDAS FILES OPEN SIMULTANEOUSLY
MAY OCCUR ONLY IF KEY IS 1. MAXIMUM NUMBER OF FILES IS 129. SEE PARAMETER MFILES IN FILE KPARAM.

DISCUSSION

1. A CALL TO NTFYMS AFTER A MIDAS FILE HAS BEEN OPENED NOTIFIES MIDAS THAT IT SHOULD LEAVE OPEN BETWEEN MIDAS CALLS ANY OF THE SPECIFIED FILE'S SEGMENT SUBFILES WHICH IT OPENS DURING SUBSEQUENT FILE ACCESS.

2. A CALL TO NTFYMS BEFORE A MIDAS FILE IS CLOSED NOTIFIES MIDAS THAT IT SHOULD CLOSE ANY OF THE FILE'S SEGMENT SUBFILES THAT IT HAS LEFT OPEN.

3. IF THE MIDAS LIBRARY HAS BEEN CUSTOMIZED TO DISABLE INTERNAL LOCKING, A CALL TO NTFYMS HAS NO EFFECT.

4. NTFYMS IS MOST USEFUL IN THOSE APPLICATIONS WHICH OPEN AND CLOSE ALL TYPES OF FILES VIA THE SAME CALLS TO THE FILE SYSTEM. IN THESE APPLICATIONS IT IS PROBABLY SIMPLER TO INSERT CALLS TO NTFYMS RATHER THAN GENERATE A SEPARATE FILE SYSTEM CALL FOR EACH TYPE OF FILE. (EG. SAM, DAM, MIDAS, ETC.)
5. NOTE THAT MIDAS DOES NOT VERIFY THAT THE FILE REFERENCED IN THE CALL TO NTFYM$ IS A MIDAS FILE. A FILE SYSTEM ERROR CODE MAY RESULT IF THE REFERENCED FILE IS NOT A MIDAS FILE.
**2.3.2.2 OPENMS**

**********
*       *
* OPENMS *
*       *
**********

**FUNCTION**

OPENMS opens a MIDAS file (segment directory) and, unless the MIDAS library has been customized to disable internal locking, causes MIDAS to leave open between MIDAS calls any of the file's segment subfiles which it opens during subsequent file access. OPENMS verifies that the specified file exists and that it is of the appropriate type, i.e. SAM segment directory.

**CALLING SEQUENCE**

CALL OPENMS (KEY, TRENAM, NAMLEN, UNIT, STATUS)

**KEY** — (INPUT) VALID SRCH$$ ACTION SUB-KEY (KSREAD, K$WRIT, OR K$RDWR, OPTIONALLY TOGETHER WITH K$GETU)

**TRENAM** — (INPUT) TREE NAME OF FILE TO BE OPENED

**NAMLEN** — (INPUT) LENGTH OF TREE NAME IN CHARACTERS

**UNIT** — (INPUT) IF K$GETU IS NOT SPECIFIED, THEN UNIT IS THE FILE UNIT ON WHICH THE FILE IS TO BE OPENED. (OUTPUT) IF K$GETU IS SPECIFIED, UNIT IS THE FILE UNIT ON WHICH THE FILE WAS OPENED.

**STATUS** — (OUTPUT) ERROR STATUS

- 0 - NO ERROR
- < 10001 - FMS ERROR (SYSTEM DEFINED)
- = 10001 - BAD KEY
- = 10002 - TOO MANY MIDAS FILES OPEN. THE LIMIT IS 129. SEE PARAMETER MFILES IN FILE KPARAM.
- = 10003 - SPECIFIED FILE IS NOT A MIDAS SEGMENT DIRECTORY
2.3.2.3 CLOSMS

**********
*        *
* CLOSMS *
*        *
**********

FUNCTION

Closes a Midas file (segment directory) open on a specified file unit and, unless the Midas library has been customized to disable internal locking, closes any of the file's segment subfiles which Midas has opened during the course of file access.

CALLING SEQUENCE

CALL CLOSMS (UNIT, CODE)

UNIT -- (INPUT) file unit on which the Midas file is open

CODE -- (OUTPUT) error status
       = 0    - no error
       > 0    - FMS error (system defined)
2.3.3 EXAMPLES

2.3.3.1 USE OF NTFYMS

IN THIS FORTRAN EXAMPLE THE PROGRAM OPENS FILE FNAME ON UNIT UNIT. VARIABLE TYPE HAS PREVIOUSLY BEEN SET TO A VALUE WHICH DESCRIBES THE TYPE OF FILE OPENED. IF THE FILE IS OF TYPE "MIDAS", THE PROGRAM CALLS NTFYMS TO NOTIFY MIDAS THAT IT IS READY TO BEGIN OPERATIONS ON THE FILE. AFTER PROCESSING HAS BEEN COMPLETED, THE PROGRAM NOTIFIES MIDAS OF THE FACT AND THEN CLOSES THE FILE. NOTE THAT NTFYMS IS USED HERE BECAUSE SEVERAL TYPES OF FILES MAY BE OPENED BY THE CALL TO SRCH$$$. NTFYMS SHOULD ONLY BE CALLED FOR MIDAS FILES.

C OPEN THE FILE
CALL SRCH$$$(K$READ,FNAME,6,UNIT,FTYPE, CODE)
IF (CODE .NE. 0) GO TO 9000
IF (TYPE .NE. MIDAS) GO TO 200/* CHECK FILE TYPE
CALL NTFYMS(1,UNIT,CODE) /* TELL MIDAS WE'RE READY
IF (CODE .NE. 0) GO TO 9002

200 CONTINUE

C DO MIDAS FILE PROCESSING (EG. CALLS TO FIND$$)

IF (TYPE .NE. MIDAS) GO TO 800
CALL NTFYMS(2,UNIT,CODE) /* TELL MIDAS PROCESSING IS DONE

800 CONTINUE
 CALL SRCH$$$(K$CLOS,0,0,UNIT,TYPE,CODE) /* CLOSE FILE
2.3.3.2 USE OF OPENMS AND CLOMS

This program uses OPENMS to open file FNAME on Unit UNIT and at the same time notify MIDAS that processing is about to begin. After processing has been completed, the program calls CLOMS to notify MIDAS that processing has been completed and to close the file. The use of OPENMS and CLOMS is convenient when one knows that only MIDAS type files are being opened or closed.

C OPEN THE FILE AND NOTIFY MIDAS THAT WE'RE READY TO USE THE FILE.
CALL OPENMS(K$READ,FNAME,6,UNIT,CODE)
IF (CODE .NE. 0) GO TO 9000

C DO MIDAS FILE PROCESSING (EG. CALLS TO FINDS)

CALL CLOMS(UNIT,CODE) /* TELL MIDAS WE'RE DONE AND CLOSE THE FILE*/
2.3.4 ADMINISTRATION CHANGES

2.3.4.1 OVERVIEW

Users must perform two types of MIDAS initialization procedures. When doing a cold start, the segment containing the lock must be shared, the lock value must be set to zero and the semaphore drained. Initialization of the semaphore and shared lock is handled by MIDAS utility IMIDAS. For details refer to section 2.3.4.2.

The second type of initialization is necessary if an application program abnormally terminates and as a consequence fails to release the shared lock. If the lock is not released, all MIDAS processes will be blocked. To release the lock, MCLUP should be executed. Note that a blocked condition might not be immediately recognized by users. If this condition is suspected, MCLUP may be executed simply to determine which process holds the lock. MCLUP is described in more detail in section 2.3.4.3.
2.3.4.2 MIDAS_INITIALIZATION -- IMIDAS

************
*        *
* IMIDAS *
*        *
************

FUNCTION

INITIALIZES THE MIDAS SEMAPHORE AND SHARED LOCK.

DISCUSSION

1. IMIDAS MUST BE RUN AS PART OF THE COLD START SEQUENCE. IF MIDAS APPLICATION PROGRAMS ARE RUNNING WHEN IMIDAS IS INVOKED, MIDAS FILES IN USE AT THE TIME MIGHT BE DAMAGED. COMMAND FILE C_MINIT MAY BE INSTALLED IN THE COLD START PROCEDURE TO SHARE THE SEGMENT CONTAINING THE LOCK AND TO EXECUTE IMIDAS.

2. IMIDAS HAS BEEN CODED AS A SUBROUTINE NAMED "MAIN" SO THAT IS CAN BE LOADED INTO SPLIT SEGMENT 4000. IMIDAS MAY THEN BE EXECUTED USING THE RESUME COMMAND.

3. COMMAND FILE C_IMIDAS IN UFD MIDAS>SOURCE MAY BE USED TO BUILD IMIDAS IN UFD MIDAS>CMDNC0.

4. IMIDAS MUST BE COMPILED WITH THE "-64V" AND "-BIG" FTN OPTIONS. DURING THE LOAD, THE COMMON BLOCK WITH THE NAME "LIST" MUST BE PLACED AT THE ADDRESS <0/1> WITH THE SEG COMMAND:

SY LIST 0 1
2.3.4.3 MIDAS CLEANUP UTILITY -- MCLUP

************
*          *
* MCLUP    *
*          *
************

FUNCTION

AFTER ABNORMAL TERMINATION OF A MIDAS PROGRAM, MCLUP RE-INITIALIZES THE SHARED LOCK AND NOTIFIES THE SEMAPHORE TO AWaken ANY MIDAS PROCESS WAITING ON THE LOCK.

DISCUSSION

1. MCLUP IS NEEDED ONLY WHEN THE ABNORMAL TERMINATION OCCURS WITHIN THE MIDAS CODE. THIS SITUATION CAN ARISE IF THE USER TYPES 'BREAK' OR 'CONTROL-P', OR IF AN INTERNAL MIDAS BUG CAUSES AN ERROR SUCH AS AN ACCESS VIOLATION.

2. IF INVOKED WITH NO OPTIONS, MCLUP RE-INITIALIZES ONLY IF THE SHARED LOCK IS HELD BY THE TERMINAL_USER. OTHERWISE MCLUP PRINTS THE USER NUMBER OF THE USER THAT HOLD THE LOCK. IF NO PROCESS HOLDS THE LOCK, THEN MCLUP DOES NOTHING.

3. IF INVOKED WITH AN OPTION OF THE FORM:

    -USER USERNUMBER

    THEN MCLUP WILL RE-INITIALIZE IF THE SHARED LOCK IS HELD BY THE SPECIFIED USER, OTHERWISE MCLUP PRINTS THE USER NUMBER OF THE USER THAT HOLDS THE LOCK. IF THE USER NUMBER OF AN ACTIVE MIDAS PROCESS IS SPECIFIED, DAMAGE MAY OCCUR TO MIDAS FILES IN USE BY THE PROCESS.

4. MCLUP MAY BE BUILT IN UFD CMDNCO BY COMMAND FILE C_MCLUP IN UFD MIDAS.

5. MCLUP MUST BE COMPILLED WITH THE "-64V" AND "-BIG" FTN OPTIONS. DURING THE LOAD, THE COMMON BLOCK WITH THE NAME "LIST" MUST BE PLACED AT THE ADDRESS <0/1> WITH THE SEG COMMAND

    SY LIST 0 1
3 RECOVERY FROM CONCURRENCY ERRORS

3.1 OVERVIEW

MIDAS NOW DETECTS AND CORRECTS MOST CONCURRENCY ERRORS. THESE ERRORS, ASSOCIATED WITH OPERATIONS INVOLVING THE CURRENT RECORD, OCCUR WHEN THE CURRENT INDEX ENTRY HAS BEEN DELETED OR PHYSICALLY MOVED SINCE THE TIME THE ENTRY BECAME CURRENT. IF MIDAS DISCOVERS THAT THE ENTRY HAS BEEN DELETED, THEN AN ERROR CODE OF 13 IS RETURNED. IN THE EVENT THAT THE ENTRY HAS BEEN MOVED, MIDAS AUTOMATICALLY LOCATES THE ENTRY AND CONTINUES NORMALLY.

3.2 IMPLEMENTATION OF CONCURRENCY ERROR DETECTION AND RECOVERY

AT THE FORTRAN CALL LEVEL INTERFACE, THE CONCEPT OF CURRENT RECORD AND CURRENT ENTRY IS IMPLEMENTED AS A FOURTEEN WORD COMMUNICATION ARRAY. THE COMMUNICATION ARRAY IS AN ARGUMENT IN MOST SUBROUTINE CALLS TO MIDAS. THE NEXT SECTION OUTLINES THE NEW COMMUNICATION ARRAY FORMAT.

3.2.1 COMMUNICATION ARRAY FORMAT

WORD 1 (INPUT) IF -1 THEN MIDAS ARRAY CONTENTS ARE NOT USED. (OUTPUT) ERROR STATUS

WORDS 2-4 CURRENT INDEX ENTRY ADDRESS
  WORD 2 BITS 1-8 — ENTRY NUMBER
  WORD 2 BITS 9-16 — SEGMENT FILE NUMBER
  WORDS 3 & 4 (32 BITS) — WORD OFFSET OF INDEX BLOCK

WORD 5 HASH VALUE (BASED ON CURRENT KEY VALUE)

WORDS 6-9 CURRENT KEY VALUE (OR 1ST 4 WORDS OF KEY)

WORDS 10-12 CURRENT RECORD ADDRESS
  WORD 10 BIT 1 — RECORD LOCKED FLAG
  WORD 10 BITS 9-16 — SEGMENT FILE NUMBER
  WORDS 11 & 12 — WORD OFFSET OF RECORD

WORD 13 DATA CONTROL WORD
  BITS 1-8 — FLAG BITS
  BITS 9-16 — PRIMARY KEY SIZE (BITS)

WORD 14 DATA RECORD LENGTH (WORDS)

NOTE THAT WORDS 2 THROUGH 9 OF THE COMMUNICATION ARRAY SPECIFY A CURRENT INDEX ENTRY AND WORDS 10 THROUGH 12 SPECIFY A CURRENT RECORD.

DURING OPERATIONS INVOLVING THE CURRENT ENTRY (E.G. GET NEXT RECORD) WORDS 2 THROUGH 4 ARE USED TO LOCATE THE EXPECTED POSITION OF THE

EVEN IF THE POINTERS DO MATCH, MIDAS COMPARES THE KEY VALUE IN THE INDEX ENTRY TO THE KEY VALUE IN THE COMMUNICATION ARRAY. IF THEY DON'T MATCH, THEN THE ENTRY IS THE WRONG ONE. WHEN A WRONG ENTRY IS DETECTED, MIDAS SEARCHES FOR THE CORRECT ENTRY. IF NOT FOUND, MIDAS RETURNS AN ERROR CODE OF 13. NOTE THAT REV 16 VERSIONS EARLIER THAN REV 16.5 RETURNED AN ERROR CODE OF 13 WHEN A CONCURRENCY ERROR WAS DETECTED. USERS OF THESE EARLIER RELEASES MAY HAVE MODIFIED THEIR APPLICATIONS TO ATTEMPT TO RECOVER FROM AN ERROR 13. AN ERROR 13 INDICATES THAT THE CURRENT INDEX ENTRY HAS BEEN DELETED, EXISTING APPLICATION ATTEMPTS TO HANDLE AN ERROR 13 MAY NEED MODIFICATION.

3.3 LIMITATIONS

FOR INDEXES WITH KEYS WHICH ARE LONGER THAN 8 BYTES, MIDAS MAY FAIL TO DETECT A CONCURRENCY ERROR. TO UNDERSTAND HOW THIS MAY OCCUR, NOTICE THAT IN THE COMMUNICATION ARRAY, AT MOST EIGHT BYTES OF A KEY MAY BE STORED. FOR KEYS LONGER THAN EIGHT BYTES, MIDAS STORES A HASH VALUE IN WORD 5 OF THE ARRAY. THE HASH VALUE IS BASED ON THE PORTION OF THE KEY BEYOND THE EIGHTH BYTE. NOW MIDAS WILL FAIL TO DETECT A CONCURRENCY ERROR IF:

A) THE DATA POINTERS MATCH (IE. THE 2 INDEX ENTRIES POINT TO THE SAME DATA RECORD),
B) THE KEY IS LONGER THAN 8 BYTES,
C) THE FIRST 8 BYTES OF THE KEY MATCH THE 8 BYTES STORED IN THE COMMUNICATION ARRAY, AND
D) THE HASH CODE, BASED ON THE REMAINING BYTES, IS THE SAME AS THE HASH CODE IN THE ARRAY.

OR IF:

A) THE DATA POINTERS MATCH,
B) THE KEYS ARE LESS THAN OR EQUAL TO 8 BYTES, AND
C) THE KEYS MATCH.
4 INSTALLATION OF MIDAS

4.1 COMMAND FILES

SEVERAL NEW COMMAND FILES HAVE BEEN ADDED.

C_MIDAS — BUILDS MIDAS LIBRARIES AND UTILITIES.

C.VKDALB — BUILDS THE SHARED V_MODE LIBRARY VKDALB.
VKDALB IS PUT IN LIB. K400Q, K2014A, AND K2014B ARE PLACED IN UFD SYSTEM.

C_NVKDALB — BUILDS THE UNSHARED V_MODE LIBRARY NVKDALB IN UFD LIB.

C_KIDALB — BUILDS THE R_MODE LIBRARY IN UFD LIB.

C.IMIDAS — BUILDS UTILTY IMIDAS IN UFD SYSTEM.

C.MCLUP — BUILDS UTILTY MCLUP IN UFD CMDNCO.

C_CREATK — BUILDS CREATK IN CMDNCO.

C_KBUILD — BUILDS KBUILD IN CMDNCO.

C_KIDDEL — BUILDS KIDDEL IN CMDNCO.

4.2 MODIFYING THE SHARED LOCK AND SEMAPHORE VALUES

AS SUPPLIED, MIDAS USES SEMAPHORE NUMBER 64 AND WORD :177777 OF SEGMENT 2020 AS THE SHARED LOCK. THESE VALUES, DEFINED IN FILE KPARAM, MAY BE MODIFIED BY USERS.

THE PARAMETERS ARE:

MSEMA1 — SEMAPHORE NUMBER
SLSEG — SEGMENT NUMBER OF THE SHARED LOCK
SLWORD — WORD NUMBER OF THE SHARED LOCK

IF ANY OF THESE VALUES IS MODIFIED, THE USER MUST FOLLOW THE PROCEDURE DESCRIBED IN PARTS 2 AND 3 OF SECTION 4.3. MIDAS UTILITIES MCLUP AND IMIDAS MUST BE REBUILT AND INSTALLED. IN ADDITION, COMMAND FILE C_MINIT AND THE COLD START PROCEDURE MUST BE MODIFIED SO THAT THE CORRECT SEGMENT GETS SHARED.

4.3 DISABLING THE NEW CONCURRENT PROCESS HANDLING METHOD

USERS MAY DISABLE THE CONCURRENCY CONTROL METHOD AND THEREBY RETURN TO THE METHOD USED IN PREVIOUS RELEASES. NOTE THAT PROGRAMS WHICH USE NTFTYM$, OPENMS$, AND CLOSMS$ WILL STILL WORK CORRECTLY.

PROCEDURE:
1) In file KPARAM, change the value of parameter SHDSEG from TRUE. to .FALSE..

2) For the unshared MIDAS libraries, KIDALB and NVKDALB,
   a) Compile subroutine LDPOOL. For V mode library
      NVKDALB use file LONGPL. For R mode library
      KIDALB use file LDPOOL.
   b) Use the binary editor, EDB, to replace the old version
      of routine LDPOOL with the new version.
   c) Reload application programs which use the unshared
      libraries.

3) For the shared V mode library VKDALB, rebuild and re-install
   the library. Application programs which use the shared
   library do not need to be re-loaded.

4.4 Network Users

For network applications in which processes access remote MIDAS
files, the concurrent process handling method must be disabled by
the user to prevent loss of file integrity.

4.5 MIDAS File Read/Write Locks

When MIDAS is installed, the read/write lock for each MIDAS file
which is to be accessed concurrently, must be set by the user to
3. (N READERS AND M WRITERS)

4.6 Reloading Application Programs

When installing MIDAS, all application programs which use an
unshared MIDAS library must be reloaded.
# TABLE OF CONTENTS

1 INTRODUCTION .................................................. 3

2 HANDLING OF CONCURRENT MIDAS PROCESSES ...................... 5
  2.1 OVERVIEW .................................................. 5
  2.2 IMPLEMENTATION METHOD .................................... 5
  2.3 APPLICATION IMPLICATIONS .................................. 6
    2.3.1 USER OPTIONS .......................................... 6
    2.3.2 APPLICATION PROGRAM MODIFICATIONS ................. 7
      2.3.2.1 NTFYMS .......................................... 8
      2.3.2.2 OPENMS .......................................... 10
      2.3.2.3 CLOSMS .......................................... 11
    2.3.3 EXAMPLES .............................................. 12
      2.3.3.1 USE OF NTFYMS ................................... 12
      2.3.3.2 USE OF OPENMS AND CLOSMS ....................... 13
    2.3.4 ADMINISTRATION CHANGES .............................. 14
      2.3.4.1 OVERVIEW ........................................ 14
      2.3.4.2 MIDAS INITIALIZATION — IMIDAS .................. 15
      2.3.4.3 MIDAS CLEANUP UTILITY — MCLUP .................. 16

3 RECOVERY FROM CONCURRENCY ERRORS .............................. 17
  3.1 OVERVIEW .................................................. 17
  3.2 IMPLEMENTATION OF CONCURRENCY ERROR DETECTION AND RECOVERY 17
    3.2.1 COMMUNICATION ARRAY FORMAT .......................... 17
  3.3 LIMITATIONS ................................................ 18

4 INSTALLATION OF MIDAS ......................................... 19
  4.1 COMMAND FILES ............................................. 19
  4.2 MODIFYING THE SHARED LOCK AND SEMAPHORE VALUES ........... 19
  4.3 DISABLING THE NEW CONCURRENT PROCESS HANDLING METHOD ...... 19
  4.4 NETWORK USERS ............................................. 20
  4.5 MIDAS FILE READ/WRITE LOCKS ............................... 20
  4.6 RELOADING APPLICATION PROGRAMS ............................ 20
MDLC1) RELEASE OF BASIC DIAGNOSTIC FOR THE 5600 (MDLC) SERIES SYNCHRONOUS CONTROLLERS

MDLC2) RELEASE OF BISYNC MICROCODE DIAGNOSTIC FOR THE 5600 (MDLC) SERIES OF SYNCHRONOUS CONTROLLERS

MDLC3) RELEASE OF PACKET MICROCODE DIAGNOSTIC FOR THE 5600 (MDLC) SERIES OF SYNCHRONOUS CONTROLLERS

MDLC4) RELEASE OF DIAGNOSTIC FOR THE ICL7020-UT200 UNIVAC 1004 MICROCODE FOR THE 5600 (MDLC) SERIES OF SYNCHRONOUS CONTROLLERS

MDLC5) RELEASE OF DIAGNOSTIC FOR HDLC MICROCODE FOR THE 5600 (MDLC) SERIES OF SYNCHRONOUS CONTROLLERS

MDLC6) RELEASE OF DIAGNOSTIC FOR BISYNC + ANY OTHER PROTOCOL ON THE 5600 (MDLC) SERIES OF SYNCHRONOUS CONTROLLERS

MDLC7) RELEASE OF DIAGNOSTIC FOR PACKET + ANY OTHER PROTOCOL ON THE 5600 (MDLC) SERIES OF SYNCHRONOUS CONTROLLERS

MDLC8) RELEASE OF DIAGNOSTIC FOR HDLC + ANY OTHER PROTOCOL ON THE 5600 (MDLC) SERIES OF SYNCHRONOUS CONTROLLERS

URCT1) TO ADD TEST FOR NEW ELECTRONIC VERTICAL FORMAT UNIT OPTION ON 1000 LPM DATA PRINTER LINE PRINTER

RTCT2) TO FIX PIO TIMING CHARACTERISTICS PERTINENT TO VCP OPERATION

HSSCT2) FAILED OCCASIONALLY ON PRIME 200'S

(COBOL) SEE MIDAS 16.5. COBOL HAS BEEN CHANGED TO WORK CORRECTLY WITH MIDAS 16.5.

(AMLCT5) TWO SMALL CHANGES WERE MADE. ONE WAS A BUG FIX AND THE OTHER IS AN ADDED FEATURE.

(FLT750) NEW TEST PROGRAM FOR P750 FLOATING POINT HARDWARE

(P500T2) TO ACCOMMODATE THE P750 CPU.
UUP146I-UP146J (CPUT4) TO ACCOMMODATE CHANGES MADE TO THE 750.

**

UUP146K-UP146L (XACHE1) TO ACCOMMODATE THE P750 CPU.

***

UUP147-UP148 (FTN) THE INTRINSIC FUNCTIONS DINT, DLOG2, AND RND WERE NOT GIVEN IMPLICIT TYPES BY THE COMPILER. THE COMPILER DID NOT DETECT ILLEGAL SPECIFICATION STATEMENTS IN V-MODE BLOCK DATA PROGRAM UNITS.

**

**

UUP149-UP150 (PRIMOS)

TAR 22486: DID NOT ALLOW THE PRIMARY PAGING SURFACE TO BE ANYTHING OTHER THAN A STORAGE MODULE.
2. A USER COULD ACCIDENTLY ASSIGN OVERLAPPING DISK PARTITIONS.
3. REMOTE LOGIN POLLING ROUTINE FIXED.
4. ATCHS$ WOULD STOP PREMATURELY IN A 3 OR MORE NODE NETWORK.
5. TAR 20005: COMINPUT COMMAND WOULD NOT PICK UP THE UNIT NUMBER CORRECTLY IF ITS POSITION WAS GREATER THAN THE THIRD TOKEN.
6. ENHANCEMENT- ALLOW USE OF ALTERNATE PAGING DEVICE.
7. ENHANCEMENT- IMPLEMENTED POLL/FINAL BIT TO PREVENT DUPLICATE PACKETS TO BE RECEIVED CAUSING LEVEL 3 RESET.

**

UUP151 (RJ2780) REV. NUMBER MESSAGE CORRECTED.

**

UUP152 (RJCDC) REV. NUMBER MESSAGE CORRECTED.

**

UUP153 (RJ1004) REV. NUMBER MESSAGE CORRECTED.

UUP154 (HASP300&400) REV. NUMBER MESSAGE CORRECTED.

**

UUP155 (BASICV)

THE FOLLOWING CHANGES HAVE BEEN MADE TO BASICV AT REV 16.6: (NOTE THAT REV 16.6 IS THE REV FOLLOWING REV 16.4. REV 16.5 WAS AN UPDATE REV TO WHICH NO CHANGE WAS MADE TO BASICV)

11. BASICV HAS BEEN MODIFIED TO TAKE ADVANTAGE OF RECENT CHANGES TO THE MIDAS PACKAGE THAT SIGNIFICANTLY IMPROVE ITS SPEED. (SEE MIDAS DOCUMENTATION FOR REV 16.5).

22. TAR 25480: IMMEDIATE-MODE STATEMENTS WITH AN ODD NUMBER OF LEADING SPACES RESULTED IN IMPROPER STATEMENT PARSING (THE LAST CHARACTER IN THE STATEMENT WAS EFFECTIVELY DROPPED). SOMETIMES THIS CAUSED WRONG ANSWERS TO CALCULATIONS. FIXED.

33. TAR 23061: THE CHANGE STATEMENT PRODUCED RANDOM RESULTS UNDER CERTAIN DIMENSIONING CONDITIONS. FIXED.

**

UUP156-UP165 (V-FTNLIB) THE FOLLOWING ROUTINES WERE REWRITTEN TO CALL
NEW (REV. 16 SHORT CALLED SCIENTIFIC FUNCTIONS.
ALOG10, ES22, TANH, ATAN2, DLOG10, ES62, ES66.
TAR 22787 - NEW VERSIONS OF SIN$X AND COS$X ARE MORE ACCURATE.
$AAD01 - WILL NO LONGER OUTPUT SINGLE SPACES BEFORE A NEWLINE.

**
UUP166-UP167 (FTNLIB) A BUG WAS FIXED IN F$10 TO FIX FREE FORMAT NUMERIC INPUT.
**
UUP168 (FAM) FIXED UNIT IN USE BUG.
**
UUP169-UP170 (LIBEDB) WILL NOW ACCEPT 64 WORD OBJECT TEXT BLOCKS
**
UUP171-UP173 (SEG) USING 'SY' COMMAND WITH NO NAME WILL MATCH FTN BLANK COMMON AGAIN.
BETTER ERROR RECOVERY ON ERRORS LIKE 'VL:'
'SY' COMMAND RESETS UNUSED WORD IN SYMBOL TABLE TO 0.
(SUPPRESSES A POSSIBLE ILLEGAL REDEFINITION MSG IF WORD NOT 0.)
**
UUP174-UP175 (LOAD) USIGN 'SY' COMMAND WITH NO NAME WILL MATCH FTN BLANK COMMON AGAIN.
WILL ABORT LOAD AFTER MSG 'CAN'T DEFER COMMON'
**
UUP176 (DBMS)
THE FOLLOWING CHANGES HAVE BEEN MADE TO REV. 16.6 DBMS:
SCHEMA - NO CHANGES.
FSUBS -
IF A SCHEMA COMPILER CONTAINED A RECORD THAT HAD NO DATA ITEMS,
CERTAIN DML COMMANDS WOULD FAIL.
CSUBS -
IF A SCHEMA CONTAINED A RECORD THAT HAD NO DATA ITEMS, CERTAIN
DML COMMANDS WOULD FAIL.
FDML - NO CHANGES.
CDML - NO CHANGES.
DMLCP -
1. THE NUMBER OF BUCKETS A RECORD OCCURRENCE CAN SPAN HAS BEEN CHANGED FROM 6 TO 20.
2. IF AN APPLICATION PROGRAM ATTEMPTS TO EXECUTE INVOKE MORE THAN ONCE WITHOUT HAVING DONE AN INTERVENING EXIT DBMS, A 1415F ERROR (INVOKE COMMAND ALREADY EXECUTED) WILL BE RETURNED TO THE USER, INSTEAD OF AN INTERNAL FATAL ERROR AS WAS IN THE PREVIOUS VERSION.
3. IF A MODIFY COMMAND WHICH NAMED AN ITEM IN A RECORD MORE THAN ONCE, AN INTERNAL FATAL ERROR WOULD RESULT.
4. UNDER CERTAIN CONDITIONS, A RECORD WITH AN ITEM OR TYPE CODE CONVERTED TO A STRING IN THE SUBSCHEMA WOULD CAUSE AN ACCESS VIOLATION.
5. IN CERTAIN CASES, DBMS RETRIEVAL COMMANDS COULD BE EXECUTED OUTSIDE OF AN ACTIVE TRANSACTION.
DBACP -
1. THE COMMANDS SAVE SCHEMA, CLEAR BEFORE, PACK, AND EXPAND NOW DO AN AUTOMATIC ROLL-BACK IF THE TRANSACTION IS LEFT OPEN.
2. THE ALGORITHM THAT COMPUTES THE BUCKET SIZE HAS BEEN CHANGED. IT WILL NOW ALLOCATE A BUCKET SIZE LARGE ENOUGH TO ACCOMODATE AN OCCURRENCE 1/3 THE SIZE OF THE LARGEST
RECORD TYPE.

3. The commands LOAD and UNLOAD have been dropped from DBACP.

4. The commands PACK, EXPAND, and ALLOCATE have been improved slightly in performance.

5. A file not open message would appear when expand of a database spanned more than one volume.

6. The after image file will not be saved if after imaging is off.

7. The performance of the clear before command has been improved.

8. The commands START AFTER, and CLEAR AFTER are now equivalent. Both commands stop the after imaging function and truncate the after image file.

9. If before image recovery is on, the command stop before will have no effect.

10. Before image recovery has been fixed so that the before image file is closed when it is completed.

11. The schema is now locked before the execution of a save after.

12. Expand files now checks to see if bucket size should be increased even if it has enough left over space in the buckets to accommodate future space needs.

SCHED-

The first release of the schema editor is available for 16.6.

**

UUP177-UP178 (FTNOPT) DINT, DLOG2, and RND are given correct implicit types now.

The compiler did not detect illegal specification statements in V-mode block data program units.

**

UUP179-UP180 (SLIST) TAR 14404- undeclared variable due to a coding error

**

UUP181-UP182 (MATHLB) TAR 80990- MATHLB, DMATHLB, IMATHLB had undeclared variables which caused loss of precision

**

UUP183-UP185 (APPLIB) TAR 13361- TIME$A was returning incorrect floating point value at the minute mark.

RNUM$A- would not accept a buffer length less than 2 characters and was not resetting negative number flag when an invalid number was passed.

**

UUP186-UP190 (SPOOL) TAR 80755 SPOOL will now use the last branch of a treename to label a request. It used to use the whole treename.

TAR 23230- SPOOL will no longer accept an AM/PM specification following a -DEFER option if the time specified is >12:59 or less than 1:00.

**

UUP191-UP192 (MAGNET) TAR 15535- odd character buffer problem fixed.
TAR 15346- INTEGER*2 VARIABLE CHANGED TO INTEGER*4.
TAR 22634- MAGNET NOW ACCEPTS UPPER OR LOWER CASE INPUT.
IN ADDITION, CALLS TO AYENAY HAVE BEEN REPLACED BY CALLS TO
APPLIB YSNOSA, THE INSERT FILE FUTCOM HAS BEEN MERGED INTO
INSERT FILE MTUCOM, TREENAMES ARE NOW ACCEPTED.

**

UUP193-UP196 (ED)

CHANGES MADE TO THE EDITOR FOR REV 16.6:
1) A PAIR OF NEW MODES CKPAR AND NCKPAR WAS ADDED. THEY
   SPECIFY WHETHER TO PRINT CHARACTERS WITH PARITY OFF AS IF
   THEY HAD PARITY ON OR TO PRINT THEM AS "NNN", WHERE N IS AN
   OCTAL NUMBER. CKPAR MEANS CHECK THE PARITY AND ONLY PRINT
   AS A REAL CHARACTER IF THE PARITY IS ON. NCKPAR MEANS
   DON'T CHECK THE PARITY -- PRINT ALL CHARACTERS SAME
   AS THEY WERE PRINTED IN REV 16.4. DEFAULT IS NCKPAR.
   (TAR 25039)
2) THE EDITOR NOW CHECKS IF BOXMODE IS ON BEFORE DECIDING
   TO PRINT THE CHARACTERS USED FOR DIRECTION IN BOXMODE AS
   "E", "N", "W", AND "S.
   (TAR 80619)
3) IF A LINE HAPPENED TO HAVE OVER 127 BLANKS ON THE
   END, THE FIRST 127 TRAILING BLANKS DID NOT GET
   TRUNCATED WHEN THE LINE WAS WRITTEN BACK TO THE FILE.

**

UUP197-UP198 (RUNOFF)

THE FOLLOWING CHANGES HAVE BEEN MADE IN RUNOFF FOR REV 16.6:
1) RUNOFF HAS BEEN MODIFIED TO ACCEPT TREENAMES UP TO
   80 CHARACTERS IN LENGTH FOR .INSERT FILES.
   (TAR 12603)
2) WHEN PROCESSING THE ".///" COMMAND FOR APPORTIONING
   TEXT ACROSS A LINE, RUNOFF
   OCCASIONALLY BLANKED
   OUT ANYTHING THAT MIGHT HAVE BEEN IN THE FIRST COLUMN
   IF THE APPORTIONED TEXT BELONGED IN A LATER COLUMN WHEN
   WORKING WITH MULTIPLE COLUMNS.
   (TAR 22802)
3) USING .+ OR .> WITHOUT FOLLOWING TEXT TO CREATE A BLANK LINE
   CAUSED RUNOFF TO PRINT TWO LINE FEEDS WHEN SENDING OUTPUT TO
   THE TERMINAL. BECAUSE THE TWO NEWLINES WERE PUT IN THE SAME WORD,
   THIS DID NOT SHOW UP WHEN THE OUTPUT WAS SENT TO A FILE AND
   SPOOLED. THE EXTRA LINES WERE NOT COUNTED AS FAR AS RUNOFF
   PAGES WERE CONCERNED EITHER WHICH CAUSED THOSE PAGES TO APPEAR
   LONGER THAN THEY ACTUALLY WERE.
   (TAR 80967)
4) THE BLANK CHARACTER DID NOT GET TRANSLATED INTO A SPACE
   WHEN IT WAS USED IN HEADERS AND FOOTERS.
5) DEFINED SYMBOLS COULD HAVE AT MOST 30 CHARACTERS BUT
   THE LOOP TO TRANSLATE THEM WENT UP TO 60 SO IF THE DEFINITION
   WAS LONGER THAN 30 THERE WAS GARBAGE ON THE END.
6) ANOTHER DEFAULT DEFINED SYMBOL TIME HAS BEEN ADDED.
   SAYING XXTIMEXX WILL BE REPLACED BY THE TIME ACCORDING TO
   THE CPU IN THE FORM HH:MM.
   S
   A RESTRICTION THAT RUNOFF USERS SHOULD BE AWARE OF IS,


WHEN USING THE TWO COMMANDS .BREAK AND .INDENT N IN CONJUNCTION 
THE .BREAK SHOULD PRECEDE THE .INDENT OR IT MAY NOT WORK AS EXPECTED.

**

UUP199-UP201

(FUTIL)

THIS IS A DOCUMENT OF THE MINOR CHANGES 
TO FUTIL FOR REVISION 16.6. THE CHANGES 
INCLUDE:
FIXING THE "TREDELETE A>B" BUG, WHERE FUTIL 
WOULDE TREDELETE A BEFORE RETURNING A SYNTAX 
ERROR.
THE UFDCOPY COMMAND CAN NO LONGER BE 
ABBREVIATED TO "U", "UF", OR "UFD".
FUTIL NOW READS IN THE COMMAND LINE USING 
COMMAND INSTEAD OF C1IN (VIA LIBRARY ROUTINE 
RDCOM), FIXING A BUG THAT CAUSED FUTIL TO BEGIN 
EXECUTION OF A COMMAND BEFORE THE RETURN KEY WAS HIT.
THE METHOD OF SIZING SAM AND DAM FILES HAS BEEN SLOWED UP.

certain error messages have become more explanatory. 
a bug causing "?NO MORE UNITS" TO BE OUTPUT AFTER 
many "INSUFFICIENT_ACCESS RIGHTS" ERRORS OCCURRED HAS BEEN FIXED.
the "TREDELETE A>B" BUG (TAR #10439) HAS BEEN FIXED.
the problem was that futil would parse the "TREDL" command, 
then parse the "A" TOKEN, DO THE "TREDELETE A", AND THEN 
TRY TO PARSE FURTHER TO DETERMINE IF THERE WERE MORE ITEMS 
TO BE TREDELETED. AT THIS POINT, FUTIL WOULD DETERMINE THAT 
THERE WAS A SYNTAX ERROR, AND REPORT IT.
FUTIL NOW PARES THE COMMAND AND THEN, IF THE COMMAND 
IS NOT A "FROM", "TO", "ATTACH", OR "*" (COMMENT) COMMAND, 
IT SCANS THE ENTIRE COMMAND LINE, LOOKING FOR A ">" SYMBOL, 
DENOTING A TREENAME. IF ONE IS FOUND, FUTIL IMMEDIATELY RETURNS 
WITH A SYNTAX ERROR.

THE "UFDCPY" COMMAND USED TO BE ABBREVIATED TO "U" (TAR 
#80855). SINCE THE COMMAND BECAME MORE POWERFUL AT REV. 14 
DUE TO THE INTRODUCTION OF THE MERGE CAPABILITY (WHICH CAN 
CAUSE FILES TO BE LOST IF MISHANDLED), IT WAS SUGGESTED IN 
THE TAR THAT "U" NO LONGER BE ACCEPTED AS AN ABBREVIATION.
MOST NEW DOCUMENTATION (NOT MANXXX DOCUMENTS) SPECIFIED 
THE MINIMUM ABBREVIATION TO BE "UFDC", ALTHOUGH AT LEAST ONE 
SPECIFIED IT AS "UFD".
FUTIL NO LONGER ACCEPTS "U", "UF", OR "UFD" AS AN ACCEPTABLE 
ABBREVIATION FOR "UFDCPY" OR FOR ANY OTHER COMMAND. IF ONE 
OF THESE ABBREVIATIONS ARE GIVEN, FUTIL WILL RESPOND WITH AN 
ERROR.
DURING INPUT, FUTIL USED TO CALL A LIBRARY SUBROUTINE NAMED 
RDCOM. THIS SUBROUTINE CALLED C1IN TO OBTAIN INPUT FROM THE TERMINAL 
AND DID ITS OWN ERASE/KILL PROCESSING. IT WOULD ALSO RETURN 
IMEDIATELY AS SOON AS 80 CHARACTERS WERE READ ;;OR== THE NEW-LINE 
CHARACTER WAS SEEN, ;;WITHOUT== GUARANTEEING THAT A NEW-LINE 
CHARACTER WAS PUT IN THE COMMAND LINE ARRAY. 
THIS MEANT THAT A) THE USER, WHILE TYPING IN A LONG COMMAND 
LINE, COULD SUDDENLY FIND THE COMMAND HAD EXECUTED BEFORE THE 
RETURN KEY HAD BEEN HIT. B) ANY COMMAND FILES THAT RAN FUTIL 
COULDN'T USE ERASE AND KILL CHARACTERS SUCCESSFULLY, WHICH
IS INCONSISTENT WITH PRIMOS AND SUBSYSTEMS THAT USE COMANL
AND RDTKS$, AND C) WHILE REPORTING SYNTAX ERRORS ON LONG COMMAND
LINES, FUTIL COULD "GO WEST" OUTPUTTING GARBAGE.

TO FIX THIS, THE CALL TO RDCOM HAS BEEN REPLACED WITH A CALL
TO COMANL FOLLOWED BY A CALL TO RDTKS$ TO READ ;;RAW TEXT== INTO
AN 80-WORD BUFFER (160 CHARACTERS) AND A DO-LOOP TO UNPACK THE
BUFFER TO RESEMBLE THE RESULTS OF CALLING RDCOM. THIS MEANS
THAT PRIMOS NOW DOES ERASE/KILL PROCESSING (AND IT WILL NOT DO
IT IF INPUT IS COMING FROM A COMMAND FILE). IT ALSO GUARANTEES THAT
HAS HIT RETURN BEFORE THE COMMAND IS EXECUTED (SINCE COMANL WILL NOT
RETURN UNTIL THEN). ALSO, IT GUARANTEES THAT A NEW-LINE CHARACTER IS
IN THE LAST WORD OF THE UNPACKED ARRAY, MEANING ALL SYNTAX
ERRORS WILL BE REPORTED ACCURATELY.

FUTIL HAS BEEN MADE SLIGHTLY FASTER WHEN
SIZING SAM AND DAM FILES, EITHER IN UFDS
OR IN SEGDIRS. FUTIL USED TO CALL PRWF$$
REPEATEDLY TO POSITION FORWARD 4096 WORDS
UNTIL AN EOF WAS FOUND, AND THEN CALL PRWF$$
TO READ THE CURRENT POSITION (WHICH REPRESENTED
THE FILE SIZE IN WORDS).

NOW, FUTIL CALLS PRWF$$ REPEATEDLY, POSITIONING
FORWARD 16,384 (2^14) WORDS IF THE FILE IS
A SAM FILE, OR 2,147,483,647 (2^31-1) WORDS IF THE
FILE IS A DAM FILE BEFORE READING THE POSITION.

THIS RESULTS IN FEWER CALLS TO PRIMOS FOR SAM
FILES (WHILE STILL BEING QUOTABLE), AND EXACTLY
2 CALLS PER FILE TO PRIMOS FOR DAM FILES (WHICH
BY THEIR NATURE CAN BE SIZED SO RAPIDLY THAT
QUITS RESPOND SPEEDILY).

THESE SPEED-UPS WILL OCCUR IN ALL
FILES OVER 4096 WORDS LONG.

ERROR MESSAGES IN FUTIL HAVE BECOME MORE EXPLANATIVE;
IN PARTICULAR, THE ERROR MESSAGE "?" FOLLOWED BY A
REPPROMPT ("">") INDICATING ONE OF THREE CONDITIONS:
A) THAT THE COMMAND WAS UNRECOGNIZABLE, B) THAT THE
COMMAND WAS AN ABBREVIATION OF ANOTHER COMMAND WHICH
IS TOO DANGEROUS TO BE ABBREVIATED, AND C) THAT A
PROTECT- OR SRWLOC-CLASS COMMAND WAS ATTEMPTED WHILE
THE FROM-DIR WAS INSIDE A SEGMENT DIRECTORY.

THE ERROR MESSAGES HAVE NOW BECOME A) "UNKNOWN COMMAND -
XXXXXX", B) "CAN'T ABBREVIATE XXXXXX COMMAND.", AND
C) "OPERATION ILLEGAL INSIDE SEGDIRS."

ALSO, A BUG IN FUTIL EXISTED WHICH CAUSED
FUTIL TO PRODUCE THE ERROR MESSAGE "NO MORE UNITS"
AFTER ENCOUNTERING MANY (APPROX. 14) ERRORS DURING
A LISTF COMMAND, EVEN THOUGH MANY UNITS WERE STILL
AVAILABLE. THIS WAS DUE TO FUTIL NOT INTERNALLY
RETURNING THE UNIT THAT IT HAD INTERNALLY ALLOCATED
AFTER ENCOUNTERING AN ERROR WHILE TRYING TO OPEN
THAT UNIT (IT REMEMBERED TO RETURN IT ONLY IF THE
ERROR WAS E$FIUS - "FILE IN USE")

FUTIL NOW REMEMBERS TO RETURN THE UNIT INTERNALLY.

**

UUP202-UP205 (COBOL)
<table>
<thead>
<tr>
<th>TAR NO.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>14219</td>
<td>SELECT WITH ALTERNATE KEY, NOT DEFINED GAVE A BAD ERROR MESSAGE.</td>
</tr>
<tr>
<td>14594</td>
<td>COMPILER DID NOT FLAG AS A ERROR A DATA-NAME WHAT WAS NOT DEFINED AS 01/77 ON A USING STATEMENT.</td>
</tr>
<tr>
<td>21498</td>
<td>MIDAS 13 / COBOL 94 STATUS CODES ALONG WITH LEAVING RECORDS LOCKED WAS FIXED WITH A NEW MIDAS AND COBOL LIBRARY WHICH IS AVAILABLE AT REV16.5.</td>
</tr>
<tr>
<td>22304</td>
<td>COMPILER FAILED TO FLAG INDEX-NAMES THAT WERE THE SAME AS DATA-NAMES.</td>
</tr>
<tr>
<td>22307</td>
<td>FIXED ON REV16.4 BAD ERROR MESSAGE WAS ISSUED WHEN NO CORRESPONDENCE WAS FOUND.</td>
</tr>
<tr>
<td>22308</td>
<td>SOME COMPILER ERROR MESSAGES HAD SPACES IN THE MIDDLE OF A WORD.</td>
</tr>
<tr>
<td>22310</td>
<td>COMPILER WAS ABORTING WITH A TABLE GROUP ERROR, IF A LEVEL 88 STATEMENT APPEARED DIRECTLY BEFORE THE LINKAGE SECTION.</td>
</tr>
<tr>
<td>22318</td>
<td>MISLEADING ERROR MESSAGE FROM COMPILER.</td>
</tr>
<tr>
<td>23360</td>
<td>MOVE CORRESPONDING WAS CAUSING THE FOLLOWING STATEMENT TO BE SKIPPED.</td>
</tr>
<tr>
<td>23677</td>
<td>COMPILER GENERATES FAULTY CODE TO EVALUATE A SUBSCRIBED NUMERIC CONDITION NAME.</td>
</tr>
<tr>
<td>23679</td>
<td>THE GENERATED CODE CONVERTED INDEX ITEMS TO COMPUTATIONAL-3, INSTEAD OF EXTERNAL DECIMAL.</td>
</tr>
<tr>
<td>23683</td>
<td>DOCUMENTATION PROBLEM, THE MANUAL DID NOT STATE THAT THE SECONDARY KEY HAD TO BE CONTAINED WITHIN THE RECORD DESCRIPTION. THE COMPILER WILL NOW FLAG THIS CONDITION.</td>
</tr>
<tr>
<td>23684</td>
<td>THE COMPILER GENERATED BAD BRANCH WHEN TESTING NEGATED CONDITION WITH A COMPUTATIONAL ITEM.</td>
</tr>
</tbody>
</table>
UUP206-UP207  80984  FIXED UNDER TAR 24806
(PXT1) TO ALLOW THE VCP TO OPERATE WITH THE
DIAGNOSTIC AS THE TEST USED TO USE THE SOC’S DIAGNOSTIC
MODE CAPABILITY WHICH ARE NOT PRESENT ON THE VCP.

**
UUP208-UP209  (AMLCT5) TWO SMALL CHANGES WERE MADE. ONE WAS A
BUG FIX AND THE OTHER IS AN ADDED FEATURE.

**
UUP210-UP212  (FLT750) NEW TEST PROGRAM FOR P750 FLOATING
POINT HARDWARE

**
UUP213  (RJECom) INITIAL REV. 16 RELEASE

**
UUP214  (RJE80) INITIAL REV. 16 RELEASE

**
UUP215  (RJE1004) INITIAL REV. 16 RELEASE

**
UUP216  (RJE200UT) INITIAL REV. 16 RELEASE

**
UUP217  (RJE7020) INITIAL REV. 16 RELEASE

**
UUP218  (RJEGRTS) INITIAL REV. 16 RELEASE

**
UUP218A  (RJEHASP) INITIAL REV. 16 RELEASE

**
UUP219-220  (COPY) ALLOW BADSPOTS ON CMD

**
UUP221-223  (APPLIB) THE ROUTINE RNUMSA WAS NOT RESETTING AN INTERNAL
NEGATIVE NUMBER FLAG ON INVALID INPUT

**
UUP224-226  (CX) TAR#20546 - CX RUNS ON SYSTEM WITH AT LEAST 16 FILE UNITS
CX DROP COMMAND NOW WORKS FOR SINGLE DIGIT (CX-D8)

**
UUP227-230  (ED & NSED) TAR#80916 - ED HUNG WHEN A * WAS NOT PRECEDED BY A
COMMAND AND WAS NOT IN THE FIRST COLUMN IN EDIT MODE
TAR#25812 - UNLOAD WILL NOW CORRECTLY COMPRESS BLANK LINES
TAR#81286 - PROBLEM WITH LINES2 HAS BEEN CORRECTED

**
UUP231-233  (MAKE) ALLOW BADSPOTS ON CMD

**
UUP234-235  (MRGF) MRGF WAS OUTPUTTING GARBAGE AT THE END OF THE
MERGED FILE

**
UUP236-237  (PRIMOS)

THIS IS A LIST OF ALL ENHANCEMENT AND TARS IN REV16.8

1. SMLC EMULATORS & NETWORKS

-----------------------------
CLEARER ERROR MESSAGES WILL BE PRINTED FOR EMULATORS AND NETWORKS
AT INITIALIZATION.
The problem of having 2 controllers configured and only one present so that the one present is still enabled is solved.

2. BAD SPOT HANDLING ON Paging Partition
---------------------------------------------
CMD BAD SPOT HANDLING ON Paging Partition.

3. SMLC For RJE
-------------
To speed up the passing of status from the SMLC driver in Ring 0 to the RJE emulators in Ring 3.

4. DPTX, 3270
-------------
To correct the interpretation of buffer addresses, so as to adhere exactly to IBM 3270 functionality. TAR # 24749

5. DPTX, 3270
-------------
Fix handling of test request function key by traffic manager. TAR # 25843

6. DPTX, 3270
-------------
To correct the handling of a device en status on a general poll TAR # 24748

**
UUP238-241 (DOS)

DOS changes for rev. 16.8

ABSTRACT
-------

1 AUTOMATIC STARTUP OF BOOT DISK
---------------------------------
DOS has been modified to perform an automatic startup of the disk from which DOS was booted. The message which DOS prints when it is first booted has been modified to appear in the form shown in the following example:
THIS CHANGE ELIMINATES THE NEED TO EXPLICITLY STARTUP THE BOOT DISK BEFORE ISSUING OTHER COMMANDS TO DOS. IF THE USER WISHES TO STARTUP A DIFFERENT PARTITION THAN THE BOOT DISK (E.G., IF HE BOOTS DISK 460 BUT WANTS 10460 STARTED UP), HE CAN STILL ISSUE THE APPROPRIATE STARTUP COMMAND TO DOS. THIS CHANGE WILL WORK FOR ALL DISK TYPES CURRENTLY SUPPORTED BY DOS.

** ADDISK SVC **

A NEW SVC HAS BEEN ADDED TO DOS TO ALLOW A PROGRAM RUNNING UNDER DOS TO ADD ADDITIONAL FILE SYSTEM PARTITIONS. THE CALLING SEQUENCE IS AS SHOWN:

```
INTEGER*2 PDEV   /* PHYSICAL DISK NUMBER
INTEGER*2 LDEV   /* LOGICAL DISK NUMBER
INTEGER*2 CODE   /* ERROR_CODE

CALL ADDISK (PDEV, LDEV, CODE)
```


NOTE THAT THE ADDISK SVC IS NOT SUPPORTED BY A FORTRAN LIBRARY INTERLUDE. IN ADDITION, ITS FUNCTIONALITY IS NOT SUPPORTED UNDER PRIMOS. IT IS INTENDED FOR USE ONLY BY THE PRIMOS PRELOADER.

**

** UUP242-243 (MDLCT5) TIME-OUT ERRORS ON P350 ON SUBTESTS 4 AND 10 **
** UUP244-245 (RTCT2) TO INCORPORATE TIMING DIFFERENCES ON THE P350 AND P750 **
** UUP246-247 (HSRPT2) TO ENABLE THE READER/PUNCH OPTION TO BE TESTED, WITH A SOC/VCP CONFIGURATION **
** UUP248-249 (XACHE1) TEST FAILED UNNECESSARILY **
** UUP250-251 (STLBT2) TEST HAD TO RUN OUT OF MACHINE CHECK MODE ON A P750 **
** UUP252-253 (P500T2) TO SHORTEN TIME FOR ONE PASS **
** UUP254-255 (CPUT4) TEST FAILED UNNECESSARILY WHEN FOLLOWING P500T2 ON A **
STRING AND WHEN RUNNING VIRTUALLY

** UUP256-257 (HSMT4) ECC TEST FAILED IF THE CPU WAS A 750 AND THE MEMORY
BOARD BEING TESTED WAS NON-INTERLEAVED

** UUP258-260 (XACHE2) TESTS THE P750'S 8K CACHE MEMORY. MEANT TO BE RUN IN
ADDITION TO XACHE1 ON THE P750

** UUP261-262 (HSMT3) MODIFICATIONS FOR EC MEMORIES AND P750.

BUG FIXES FOR CACHE DISABLE.