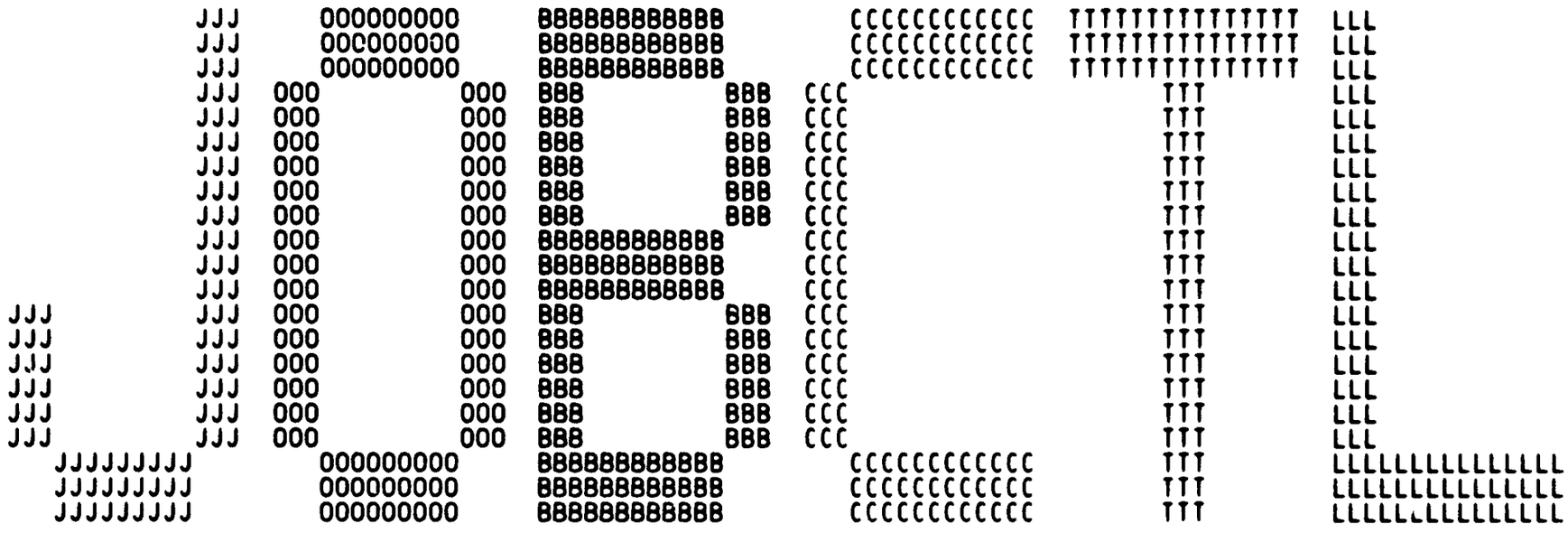

_$2

Val

0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
0000
314
485
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FFF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF

FILEID**BATCH

BA
VO4

```
BBBBBBBB       AAAAAA    TTTTTTTTTT    CCCCCCCC  HH       HH
BBBBBBBB       AAAAAA    TTTTTTTTTT    CCCCCCCC  HH       HH
BB       BB   AA      AA       TT      CC        HH       HH
BB       BB   AA      AA       TT      CC        HH       HH
BB       BB   AA      AA       TT      CC        HH       HH
BB       BB   AA      AA       TT      CC        HH       HH
BBBBBBBB      AA      AA       TT      CC        HHHHHHHHHH
BBBBBBBB      AA      AA       TT      CC        HHHHHHHHHH
BB       BB   AAAAAAAAAA       TT      CC        HH       HH
BB       BB   AAAAAAAAAA       TT      CC        HH       HH
BB       BB   AA      AA       TT      CC        HH       HH      ....
BB       BB   AA      AA       TT      CC        HH       HH      ....
BBBBBBBB      AA      AA       TT        CCCCCCCC  HH       HH      ....
BBBBBBBB      AA      AA       TT        CCCCCCCC  HH       HH      ....

LL              IIIIII      SSSSSSSS
LL              IIIIII      SSSSSSSS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II      SS
LL                II        SSSSSS
LL                II        SSSSSS
LL                II              SS
LL                II              SS
LL                II              SS
LL                II              SS
LLLLLLLLLL      IIIIII    SSSSSSSS
LLLLLLLLLL      IIIIII    SSSSSSSS
```

```
;     1        0001  0 MODULE BATCH    (%TITLE 'Batch process control'
;     2        0002  0                  IDENT = 'V04-000'
;     3        0003  0                  ) =
;     4        0004  1 BEGIN
;     5        0005  1
;     6        0006  1 !
;     7        0007  1 !**************************************************************************
;     8        0008  1 !*                                                                        *
;     9        0009  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;    10        0010  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;    11        0011  1 !*  ALL RIGHTS RESERVED.                                                  *
;    12        0012  1 !*                                                                        *
;    13        0013  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    14        0014  1 !*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE        *
;    15        0015  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER   *
;    16        0016  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
;    17        0017  1 !*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY     *
;    18        0018  1 !*  TRANSFERRED.                                                          *
;    19        0019  1 !*                                                                        *
;    20        0020  1 !*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE  *
;    21        0021  1 !*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT      *
;    22        0022  1 !*  CORPORATION.                                                          *
;    23        0023  1 !*                                                                        *
;    24        0024  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS  *
;    25        0025  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.               *
;    26        0026  1 !*                                                                        *
;    27        0027  1 !*                                                                        *
;    28        0028  1 !**************************************************************************
;    29        0029  1 !
;    30        0030  1
;    31        0031  1 !++
;    32        0032  1 ! FACILITY:
;    33        0033  1 !       Job controller.
;    34        0034  1 !
;    35        0035  1 ! ABSTRACT:
;    36        0036  1 !       This module contains the routines specific to batch processing.
;    37        0037  1 !
;    38        0038  1 ! ENVIRONMENT:
;    39        0039  1 !       VAX/VMS user and kernel mode.
;    40        0040  1 !--
;    41        0041  1 !
;    42        0042  1 ! AUTHOR: M. Jack, CREATION DATE: 16-Feb-1982
;    43        0043  1 !
;    44        0044  1 ! MODIFIED BY:
;    45        0045  1 !
;    46        0046  1 !       V03-006 KPL0001         P Lieberwirth,   9-Jul-1984
;    47        0047  1 !               Eliminate a source of queue file corruption in routine
;    48        0048  1 !               BATCH_DELETION.  Specifically, if the SJH describing a
;    49        0049  1 !               batch job being deleted was deallocated to the free list
;    50        0050  1 !               by the routine COMPLETE_JOB, and then a crash occured
;    51        0051  1 !               before routine BATCH_DELETION could finish the operation
;    52        0052  1 !               by re-writing the SMQ, the queue file would contain the
;    53        0053  1 !               old SMQ record image which now contained a pointer to a
;    54        0054  1 !               record on the free list.  Other routines in the JOB
;    55        0055  1 !               CONTROLLER would trip over this corruption, generally by
;    56        0056  1 !               trying to to follow a zero pointer in the now deallocated-SJH
;    57        0057  1 !               and encountering an RMS invalid-key by trying to read record
```

```
;    58          0058  1 !                zero.
;    59          0059  1 !
;    60          0060  1 !                The fix is to flush the SMQ before doing the complete job.
;    61          0061  1 !                This results in an extra read operation, since the SMQ is
;    62          0062  1 !                needed again after COMPLETE_JOB returns.  However, the extra
;    63          0063  1 !                trip to read_record is not so expensive because the SMQ
;    64          0064  1 !                may still have a non-zero reference count and as a result
;    65          0065  1 !                still be in the cache.  At any rate, the extra trip avoids
;    66          0066  1 !                possible file corruption.
;    67          0067  1 !
;    68          0068  1 !                By flushing the SMQ before doing COMPLETE_JOB on the SJH,
;    69          0069  1 !                we traded a window where if a crash occured file corruption
;    70          0070  1 !                would result, for a window where if a crash occured, we lost
;    71          0071  1 !                a record describing a batch job that was to be deleted.  The
;    72          0072  1 !                trade is a good one.
;    73          0073  1 !
;    74          0074  1 !        V03-005 PCG0001         Peter George    27-Feb-1984
;    75          0075  1 !                Fix CPU time limit logic.
;    76          0076  1 !
;    77          0077  1 !        V03-004 MLJ0115         Martin L. Jack, 30-Jul-1983  14:33
;    78          0078  1 !                Changes for job controller baselevel.
;    79          0079  1 !
;    80          0080  1 !        V03-003 MLJ0114         Martin L. Jack, 23-Jun-1983  4:56
;    81          0081  1 !                Changes for job controller baselevel.
;    82          0082  1 !
;    83          0083  1 !        V03-002 MLJ0113         Martin L. Jack, 26-May-1983  21:06
;    84          0084  1 !                Changes for job controller baselevel.
;    85          0085  1 !
;    86          0086  1 !        V03-001 MLJ0112         Martin L. Jack, 29-Apr-1983  2:52
;    87          0087  1 !                Changes for job controller baselevel.
;    88          0088  1 !
;    89          0089  1 !**
```

```
;    91          0090  1 REQUIRE 'SRC$:JOBCTLDEF';
;    92          1131  1
;    93          1132  1
;    94          1133  1 FORWARD ROUTINE
;    95          1134  1         SJC_BATCH_SERVICE,
;    96          1135  1         BATCH_DELETION:                 NOVALUE;
;    97          1136  1
;    98          1137  1
;    99          1138  1 EXTERNAL ROUTINE
;   100          1139  1         COMPLETE_JOB:                   NOVALUE,
;   101          1140  1         COMPLETE_SRB_OUTPUT_ITEM:       NOVALUE,
;   102          1141  1         CREATE_SRB:                     NOVALUE,
;   103          1142  1         FETCH_VARIABLE_ITEM,
;   104          1143  1         FETCH_VARIABLE_ITEM_LIST,
;   105          1144  1         FIND_PENDING_JOBS:              NOVALUE,
;   106          1145  1         FIND_PROCESS_DATA:              L_OUTPUT_3,
;   107          1146  1         FLUSH_RECORD:                   NOVALUE,
;   108          1147  1         LOCATE_SRB_OUTPUT_ITEM,
;   109          1148  1         READ_RECORD,
;   110          1149  1         RELEASE_RECORD:                 NOVALUE,
;   111          1150  1         REWRITE_RECORD:                 NOVALUE,
;   112          1151  1         SEND_SERVICE_RESPONSE_MESSAGE:  NOVALUE,
;   113          1152  1         UPDATE_GETQUI_DATA:             NOVALUE;
;   114          1153  1
;   115          1154  1
;   116          1155  1 BUILTIN
;   117          1156  1         MOVC3,
;   118          1157  1         MOVC5;
```

```
;  120          1158  1 GLOBAL ROUTINE SJC_BATCH_SERVICE=
;  121          1159  1
;  122          1160  1 !++
;  123          1161  1 !
;  124          1162  1 ! FUNCTIONAL DESCRIPTION:
;  125          1163  1 !       This routine processes the SJC$_BATCH_SERVICE request.
;  126          1164  1 !
;  127          1165  1 !  INPUT PARAMETERS:
;  128          1166  1 !       NONE
;  129          1167  1 !
;  130          1168  1 ! IMPLICIT INPUTS:
;  131          1169  1 !       MBX             - Pointer to buffered mailbox message.
;  132          1170  1 !
;  133          1171  1 ! OUTPUT PARAMETERS:
;  134          1172  1 !       NONE
;  135          1173  1 !
;  136          1174  1 ! IMPLICIT OUTPUTS:
;  137          1175  1 !       NONE
;  138          1176  1 !
;  139          1177  1 ! ROUTINE VALUE:
;  140          1178  1 !       Completion status to be returned to requestor.
;  141          1179  1 !
;  142          1180  1 ! SIDE EFFECTS:
;  143          1181  1 !       NONE
;  144          1182  1 !
;  145          1183  1 !--
;  146          1184  1
;  147          1185  2 BEGIN
;  148          1186  2 LOCAL
;  149          1187  2         SJH_N,                          ! Record number of SJH
;  150          1188  2         SJH:            REF BBLOCK,     ! Pointer to SJH
;  151          1189  2         SMQ_N,                          ! Record number of SMQ
;  152          1190  2         SMQ:            REF BBLOCK,     ! Pointer to SMQ
;  153          1191  2         SQR_N,                          ! Record number of SQR
;  154          1192  2         SQR:            REF BBLOCK,     ! Pointer to SQR
;  155          1193  2         DJI:            REF BBLOCK,     ! Base of DJI item list
;  156          1194  2         DJIITM:         REF BBLOCK,     ! Cursor for DJI item list
;  157          1195  2         DJIFLG:         REF BBLOCK,     ! Pointer to DJI flags longword
;  158          1196  2         SRB:            BBLOCK[1024],   ! Local SRB
;  159          1197  2         FLAGS:          BBLOCK[4],      ! Local INPUT_FLAGS
;  160          1198  2         T;                              ! Temporary for quota calculations
;  161          1199  2
;  162          1200  2
;  163          1201  2 ! Ensure that the requesting process has CMKRNL privilege.
;  164          1202  2 !
;  165          1203  2 IF NOT .BBLOCK[MBX[ACM$Q_PRVMSK], PRV$V_CMKRNL]
;  166          1204  2 THEN
;  167          1205  2     RETURN JBC$_NOCMKRNL;
;  168          1206  2
;  169          1207  2
;  170          1208  2 ! Locate the data for this job.
;  171          1209  2 !
;  172          1210  2 IF NOT FIND_PROCESS_DATA(
;  173          1211  2     PDE_K_BATCH, .MBX[ACM$L_PID], FALSE;
;  174          1212  2     , SMQ_N, SJH_N)
;  175          1213  2 THEN
;  176          1214  2     RETURN JBC$_NOSUCHJOB;
```

```
;  177     1215  2
;  178     1216  2
;  179     1217  2 ! Read the queue record and the job record.
;  180     1218  2 !
;  181     1219  2 SMQ = READ_RECORD(.SMQ_N);
;  182     1220  2 SJH = READ_RECORD(.SJH_N);
;  183     1221  2
;  184     1222  2
;  185     1223  2 ! Scan the input item buffer, if specified.
;  186     1224  2 !
;  187     1225  2 FLAGS = 0;
;  188     1226  2 IF .ITEM_PRESENT[SJC$_BATCH_INPUT]
;  189     1227  2 THEN
;  190     1228  3     BEGIN
;  191     1229  3     LOCAL
;  192     1230  3         P:                  REF BBLOCK,     ! Cursor for item list
;  193     1231  3         P_END;                              ! Pointer past end of item list
;  194     1232  3
;  195     1233  3
;  196     1234  3     ! Pick up a pointer to the item list and one to the last item.
;  197     1235  3     !
;  198     1236  3     P = .VALUE_BATCH_INPUT[SDSC_A_POINTER];
;  199     1237  3     P_END = .P + .VALUE_BATCH_INPUT[SDSC_W_LENGTH] - 4;
;  200     1238  3
;  201     1239  3
;  202     1240  3     ! Loop over the items.
;  203     1241  3     !
;  204     1242  3     WHILE .P LSSA .P_END DO
;  205     1243  4         BEGIN
;  206     1244  4         LOCAL
;  207     1245  4             TYPE,                           ! Item type
;  208     1246  4             SIZE;                           ! Item size
;  209     1247  4
;  210     1248  4
;  211     1249  4         ! Get and advance over the item type and size.
;  212     1250  4         !
;  213     1251  4         TYPE = .P[DJI$W_ITEM_CODE];
;  214     1252  4         SIZE = .P[DJI$W_ITEM_SIZE];
;  215     1253  4         P = .P + DJI$S_ITEM_HEADER;
;  216     1254  4
;  217     1255  4
;  218     1256  4         ! Process the item.
;  219     1257  4         !
;  220     1258  4         CASE .TYPE FROM DJI$K_INPUT_FLAGS TO DJI$K_CONDITION_VECTOR OF
;  221     1259  4             SET
;  222     1260  4
;  223     1261  4
;  224     1262  4             [OUTRANGE]:
;  225     1263  4                 EXITLOOP;
;  226     1264  4
;  227     1265  4
;  228     1266  4             [DJI$K_INPUT_FLAGS]:
;  229     1267  5                 BEGIN
;  230     1268  5                 IF .SIZE EQL 4
;  231     1269  5                 THEN
;  232     1270  5                     FLAGS = ..P;
;  233     1271  4                 END;
```

```
;  234     1272  4
;  235     1273  4
;  236     1274  4              [DJI$K_CONDITION_VECTOR]:
;  237     1275  5                  BEGIN
;  238     1276  5                  IF .SIZE LEQU 12
;  239     1277  5                  THEN
;  240     1278  5                      MOVC5(
;  241     1279  5                          SIZE, .P,
;  242     1280  5                          %REF(0),
;  243     1281  5                          %REF(SJH$S_CONDITION_VECTOR), SJH[SJH$L_CONDITION_1]);
;  244     1282  4                  END;
;  245     1283  4
;  246     1284  4
;  247     1285  4              TES;
;  248     1286  4
;  249     1287  4
;  250     1288  4          ! Advance to the next item.
;  251     1289  4          !
;  252     1290  4          P = .P + .SIZE;
;  253     1291  3          END;
;  254     1292  2      END;
;  255     1293  2
;  256     1294  2
;  257     1295  2 ! Initialize the SRB.
;  258     1296  2 !
;  259     1297  2 CREATE_SRB(SRB);
;  260     1298  2 DJIITM = DJI = LOCATE_SRB_OUTPUT_ITEM(
;  261     1299  2      SRB,
;  262     1300  2      SJC$_BATCH_OUTPUT, VALUE_BATCH_OUTPUT);
;  263     1301  2
;  264     1302  2
;  265     1303  2 IF .DJIITM NEQ 0
;  266     1304  2 THEN
;  267     1305  3      BEGIN
;  268     1306  3
;  269     1307  3      ! Begin the DJI item list.
;  270     1308  3      !
;  271     1309  3      DJIITM[DJI$W_ITEM_SIZE] = DJI$S_FLAGS;
;  272     1310  3      DJIITM[DJI$W_ITEM_CODE] = DJI$K_FLAGS;
;  273     1311  3      DJIFLG = DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  274     1312  3      DJIITM[DJI$L_FLAGS] = 0;
;  275     1313  3      DJIITM = .DJIITM + DJI$S_FLAGS;
;  276     1314  3      DJIFLG[DJI$V_TERMINATE] = TRUE;
;  277     1315  3
;  278     1316  3
;  279     1317  3      ! Flags.
;  280     1318  3      !
;  281     1319  3      IF .SJH[SJH$V_NOTIFY] THEN DJIFLG[DJI$V_NOTIFY] = TRUE;
;  282     1320  3      IF .SJH[SJH$V_RESTARTING] THEN DJIFLG[DJI$V_RESTARTING] = TRUE;
;  283     1321  3      IF .SJH[SJH$V_LOG_NULL]
;  284     1322  3      THEN
;  285     1323  3          DJIFLG[DJI$V_LOG_NULL] = TRUE
;  286     1324  3      ELSE
;  287     1325  4          BEGIN
;  288     1326  4          IF .SJH[SJH$V_LOG_DELETE] THEN DJIFLG[DJI$V_LOG_DELETE] = TRUE;
;  289     1327  4          IF .SJH[SJH$V_LOG_SPOOL] THEN DJIFLG[DJI$V_LOG_SPOOL] = TRUE;
;  290     1328  3          END;
```

```
;  291          1329  3
;  292          1330  3
;  293          1331  3      ! Checkpoint data.
;  294          1332  3      !
;  295          1333  3      DJIITM = FETCH_VARIABLE_ITEM(
;  296          1334  3          SJH$S_CHECKPOINT, SJH[SJH$T_CHECKPOINT],
;  297          1335  3          DJI$K_RESTART,
;  298          1336  3          .DJIITM);
;  299          1337  3
;  300          1338  3
;  301          1339  3      ! CPU maximum.
;  302          1340  3      !
;  303          1341  3      T = 0;
;  304          1342  3      IF .SJH[SJH$V_CPU_MAXIMUM] THEN T = .SJH[SJH$L_CPU_MAXIMUM]
;  305          1343  3      ELSE IF .SMQ[SMQ$V_CPU_DEFAULT] THEN T = .SMQ[SMQ$L_CPU_DEFAULT];
;  306          1344  3      IF .SMQ[SMQ$V_CPU_MAXIMUM]
;  307          1345  3      THEN
;  308          1346  4          BEGIN
;  309          1347  4          DJIFLG[DJI$V_USE_CPU_MAXIMUM] = TRUE;
;  310          1348  4          IF .SMQ[SMQ$L_CPU_MAXIMUM] - 1 LSSU .T - 1
;  311          1349  4          THEN
;  312          1350  4              T = .SMQ[SMQ$L_CPU_MAXIMUM];
;  313          1351  3          END;
;  314          1352  3      IF .SJH[SJH$V_CPU_MAXIMUM]
;  315          1353  3      OR .SMQ[SMQ$V_CPU_DEFAULT]
;  316          1354  3      OR .SMQ[SMQ$V_CPU_MAXIMUM]
;  317          1355  3      THEN
;  318          1356  4          BEGIN
;  319          1357  4          DJIITM[DJI$W_ITEM_SIZE] = 4;
;  320          1358  4          DJIITM[DJI$W_ITEM_CODE] = DJI$K_CPU_MAXIMUM;
;  321          1359  4          DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  322          1360  4          .DJIITM = .T;
;  323          1361  4          DJIITM = .DJIITM + 4;
;  324          1362  3          END;
;  325          1363  3
;  326          1364  3
;  327          1365  3      ! Job name.
;  328          1366  3      !
;  329          1367  3      DJIITM[DJI$W_ITEM_SIZE] = CH$RCHAR(SJH[SJH$T_NAME]);
;  330          1368  3      DJIITM[DJI$W_ITEM_CODE] = DJI$K_JOB_NAME;
;  331          1369  3      DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  332          1370  3      MOVC3(
;  333          1371  3          %REF(CH$RCHAR(SJH[SJH$T_NAME])),
;  334          1372  3          SJH[SJH$T_NAME] + 1,
;  335          1373  3          .DJIITM; ,,, DJIITM);
;  336          1374  3
;  337          1375  3
;  338          1376  3      ! Log file queue.
;  339          1377  3      !
;  340          1378  3      IF .SJH[SJH$L_LOG_QUEUE_LINK] NEQ 0
;  341          1379  3      THEN
;  342          1380  4          BEGIN
;  343          1381  4          LOCAL
;  344          1382  4              SMQ_N2,                         ! Record number of log SMQ
;  345          1383  4              SMQ_2:              REF BBLOCK;     ! Pointer to log SMQ
;  346          1384  4
;  347          1385  4          SMQ_2 = READ_RECORD(SMQ_N2 = .SJH[SJH$L_LOG_QUEUE_LINK]);
```

```
;  348          1386  4          DJIITM[DJI$W_ITEM_SIZE] = CH$RCHAR(SMQ_2[SMQ$T_NAME]);
;  349          1387  4          DJIITM[DJI$W_ITEM_CODE] = DJI$K_LOG_QUEUE;
;  350          1388  4          DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  351          1389  4          MOVC3(
;  352          1390  4              %REF(CH$RCHAR(SMQ_2[SMQ$T_NAME])),
;  353          1391  4              SMQ_2[SMQ$T_NAME] + 1,
;  354          1392  4              .DJIITM; ,,, DJIITM);
;  355          1393  4          RELEASE_RECORD(.SMQ_N2);
;  356          1394  3          END;
;  357          1395  3
;  358          1396  3
;  359          1397  3      ! Log file specification.
;  360          1398  3      !
;  361          1399  3      DJIITM = FETCH_VARIABLE_ITEM(
;  362          1400  3          SJH$S_LOG_SPECIFICATION, SJH[SJH$T_LOG_SPECIFICATION],
;  363          1401  3          DJI$K_LOG_SPECIFICATION,
;  364          1402  3          .DJIITM);
;  365          1403  3
;  366          1404  3
;  367          1405  3      ! Parameters.
;  368          1406  3      !
;  369          1407  3      DJIITM = FETCH_VARIABLE_ITEM_LIST(
;  370          1408  3          SJH$S_PARAMETERS, SJH[SJH$T_PARAMETERS],
;  371          1409  3          DJI$K_PARAMETER_1,
;  372          1410  3          .DJIITM);
;  373          1411  3
;  374          1412  3
;  375          1413  3      ! User name.
;  376          1414  3      !
;  377          1415  3      DJIITM[DJI$W_ITEM_SIZE] = SJH$S_USERNAME;
;  378          1416  3      DJIITM[DJI$W_ITEM_CODE] = DJI$K_USERNAME;
;  379          1417  3      DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  380          1418  3      MOVC3(
;  381          1419  3          %REF(SJH$S_USERNAME),
;  382          1420  3          SJH[SJH$T_USERNAME],
;  383          1421  3          .DJIITM; ,,, DJIITM);
;  384          1422  3
;  385          1423  3
;  386          1424  3      ! Working set default.
;  387          1425  3      !
;  388          1426  3      T = -1;
;  389          1427  3      IF .SMQ[SMQ$V_WSDEFAULT]
;  390          1428  3      THEN
;  391          1429  4          BEGIN
;  392          1430  4          DJIFLG[DJI$V_USE_WSDEFAULT] = TRUE;
;  393          1431  4          T = .SMQ[SMQ$W_WSDEFAULT];
;  394          1432  3          END;
;  395          1433  3      IF .SJH[SJH$V_WSDEFAULT]
;  396          1434  3      THEN
;  397          1435  4          BEGIN
;  398          1436  4          IF .SJH[SJH$W_WSDEFAULT] LSSU .T THEN T = .SJH[SJH$W_WSDEFAULT];
;  399          1437  3          END;
;  400          1438  3      IF .T GEQ 0
;  401          1439  3      THEN
;  402          1440  4          BEGIN
;  403          1441  4          DJIITM[DJI$W_ITEM_SIZE] = 4;
;  404          1442  4          DJIITM[DJI$W_ITEM_CODE] = DJI$K_WSDEFAULT;
```

```
;  405     1443  4          DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  406     1444  4          .DJIITM = .T;
;  407     1445  4          DJIITM = .DJIITM + 4;
;  408     1446  3          END;
;  409     1447  3
;  410     1448  3
;  411     1449  3      ! Working set extent.
;  412     1450  3      !
;  413     1451  3      T = -1;
;  414     1452  3      IF .SMQ[SMQ$V_WSEXTENT]
;  415     1453  3      THEN
;  416     1454  4          BEGIN
;  417     1455  4          DJIFLG[DJI$V_USE_WSEXTENT] = TRUE;
;  418     1456  4          T = .SMQ[SMQ$W_WSEXTENT];
;  419     1457  3          END;
;  420     1458  3      IF .SJH[SJH$V_WSEXTENT]
;  421     1459  3      THEN
;  422     1460  4          BEGIN
;  423     1461  4          IF .SJH[SJH$W_WSEXTENT] LSSU .T THEN T = .SJH[SJH$W_WSEXTENT];
;  424     1462  3          END;
;  425     1463  3      IF .T GEQ 0
;  426     1464  3      THEN
;  427     1465  4          BEGIN
;  428     1466  4          DJIITM[DJI$W_ITEM_SIZE] = 4;
;  429     1467  4          DJIITM[DJI$W_ITEM_CODE] = DJI$K_WSEXTENT;
;  430     1468  4          DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  431     1469  4          .DJIITM = .T;
;  432     1470  4          DJIITM = .DJIITM + 4;
;  433     1471  3          END;
;  434     1472  3
;  435     1473  3
;  436     1474  3      ! Working set quota.
;  437     1475  3      !
;  438     1476  3      T = -1;
;  439     1477  3      IF .SMQ[SMQ$V_WSQUOTA]
;  440     1478  3      THEN
;  441     1479  4          BEGIN
;  442     1480  4          DJIFLG[DJI$V_USE_WSQUOTA] = TRUE;
;  443     1481  4          T = .SMQ[SMQ$W_WSQUOTA];
;  444     1482  3          END;
;  445     1483  3      IF .SJH[SJH$V_WSQUOTA]
;  446     1484  3      THEN
;  447     1485  4          BEGIN
;  448     1486  4          IF .SJH[SJH$W_WSQUOTA] LSSU .T THEN T = .SJH[SJH$W_WSQUOTA];
;  449     1487  3          END;
;  450     1488  3      IF .T GEQ 0
;  451     1489  3      THEN
;  452     1490  4          BEGIN
;  453     1491  4          DJIITM[DJI$W_ITEM_SIZE] = 4;
;  454     1492  4          DJIITM[DJI$W_ITEM_CODE] = DJI$K_WSQUOTA;
;  455     1493  4          DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  456     1494  4          .DJIITM = .T;
;  457     1495  4          DJIITM = .DJIITM + 4;
;  458     1496  3          END;
;  459     1497  3
;  460     1498  3
;  461     1499  3      IF NOT .FLAGS[DJI$V_NO_FILE]
```

```
;  462     1500  3      THEN
;  463     1501  4          BEGIN
;  464     1502  4
;  465     1503  4          ! Locate the first or next file in the job.
;  466     1504  4          !
;  467     1505  4          IF .SJH[SJH$L_CURRENT_FILE_LINK] EQL 0
;  468     1506  4          THEN
;  469     1507  4              SQR_N = .SJH[SJH$L_FILE_LIST]
;  470     1508  4          ELSE
;  471     1509  5              BEGIN
;  472     1510  5              SQR = READ_RECORD(.SJH[SJH$L_CURRENT_FILE_LINK]);
;  473     1511  5              SQR_N = .SQR[SYM$L_LINK];
;  474     1512  5              RELEASE_RECORD(.SJH[SJH$L_CURRENT_FILE_LINK]);
;  475     1513  4              END;
;  476     1514  4
;  477     1515  4
;  478     1516  4          ! Update the current file link.
;  479     1517  4          !
;  480     1518  4          SJH[SJH$L_CURRENT_FILE_LINK] = .SQR_N;
;  481     1519  4
;  482     1520  4
;  483     1521  4          ! If the job is not complete, pass the next file to the job.
;  484     1522  4          !
;  485     1523  4          IF .SQR_N NEQ 0
;  486     1524  4          THEN
;  487     1525  5              BEGIN
;  488     1526  5
;  489     1527  5              ! Read the SQR record.
;  490     1528  5              !
;  491     1529  5              SQR = READ_RECORD(.SQR_N);
;  492     1530  5
;  493     1531  5
;  494     1532  5              ! Flags.
;  495     1533  5              !
;  496     1534  5              DJIFLG[DJI$V_TERMINATE] = FALSE;
;  497     1535  5
;  498     1536  5
;  499     1537  5              ! Command file ID.
;  500     1538  5              !
;  501     1539  5              DJIITM[DJI$W_ITEM_SIZE] = SQR$S_FILE_IDENTIFICATION;
;  502     1540  5              DJIITM[DJI$W_ITEM_CODE] = DJI$K_FILE_IDENTIFICATION;
;  503     1541  5              DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  504     1542  5              MOVC3(
;  505     1543  5                  %REF(SQR$S_FILE_IDENTIFICATION),
;  506     1544  5                  SQR[SQR$T_FILE_IDENTIFICATION],
;  507     1545  5                  .DJIITM; ,,, DJIITM);
;  508     1546  5
;  509     1547  5
;  510     1548  5              RELEASE_RECORD(.SQR_N);
;  511     1549  4              END;
;  512     1550  3          END;
;  513     1551  3
;  514     1552  3
;  515     1553  3      ! Terminate the item list.
;  516     1554  3      !
;  517     1555  3      DJIITM[DJI$W_ITEM_SIZE] = 0;
;  518     1556  3      DJIITM[DJI$W_ITEM_CODE] = 0;
```

```
;  519          1557  3     DJIITM = .DJIITM + DJI$S_ITEM_HEADER;
;  520          1558  3
;  521          1559  3
;  522          1560  3     COMPLETE_SRB_OUTPUT_ITEM(
;  523          1561  3         SRB,
;  524          1562  3         .DJIITM - .DJI);
;  525          1563  2     END;
;  526          1564  2
;  527          1565  2
;  528          1566  2 ! Rewrite the job header.
;  529          1567  2 !
;  530          1568  2 REWRITE_RECORD(.SJH_N);
;  531          1569  2
;  532          1570  2
;  533          1571  2 ! Send the resporse message locally and then return a status of zero to inhibit
;  534          1572  2 ! the central response return.
;  535          1573  2 !
;  536          1574  2 SEND_SERVICE_RESPONSE_MESSAGE(SRB, SS$_NORMAL);
;  537          1575  2 0
;  538          1576  1 END;


                                .TITLE  BATCH Batch process control
                                .IDENT  \V04-000\

                                .PSECT  COMMON,NOEXE,  OVR,2

                        00000 DIAG_STORAGE_BASE:
                                .BLKB   0
                        00000 DIAG_TRACE:
                                .BLKB   96
                        00060 DIAG_COUNT:
                                .BLKB   96
                        000C0 DIAG_FLAGS:
                                .BLKB   4
                        000C4 WORK_AREA:
                                .BLKB   44
                        000F0 SNDJBC_COUNT:
                                .BLKB   132
                        00174 GETQUI_COUNT:
                                .BLKB   40
                        0019C SNDACC_COUNT:
                                .BLKB   28
                        001B8 SNDSMB_COUNT:
                                .BLKB   72
                        00200 DIAG_STORAGE_END:
                                .BLKB   0
                        00200 FLAGS:    .BLKB   4
                        00204 IMAGE_DUMP_STSFLG:
                                .BLKB   4
                        00208 THIS_SYSID:
                                .BLKB   6
                        0020E           .BLKB   2
                        00210 CUR_TIME:
                                .BLKB   8
                        00218 HOURLY_TIME:
                                .BLKB   8
```

```
00220 HOURLY_PARAMS:
                .BLKB   20
00234 SYMBIONT_COUNT:
                .BLKB   4
00238 QUEUE_REFERENCE_COUNT:
                .BLKB   4
0023C MBX_MESSAGE_COUNT:
                .BLKB   4
00240 MBX:      .BLKB   4
00244 MBX_END:  .BLKB   4
00248 MEMORY_FREE_QUEUES:
                .BLKB   40
00270 NONAST_WORK_QUEUE:
                .BLKB   8
00278 BCB_FREE_LIST:
                .BLKB   4
0027C BCB_ACTIVE_LIST:
                .BLKB   4
00280 GQL_FREE_LIST:
                .BLKB   4
00284 GQL_ACTIVE_LIST:
                .BLKB   4
00288 OPEN_GETQUI_LIST:
                .BLKB   4
0028C PROCESS_DATA_LIST:
                .BLKB   4
00290 SYMBIONT_CONTROL:
                .BLKB   4
00294 SPARE_AREA:
                .BLKB   12
002A0 REMOTE_REQUEST_LKSB:
                .BLKB   8
002A8 QUEUE_FILE_LKSB:
                .BLKB   8
002B0 QUEUE_LOCK_LKSB:
                .BLKB   8
002B8 RSP:      .BLKB   8
002C0 JBC_PRIORITY:
                .BLKB   4
002C4 JBC_PRIVILEGES:
                .BLKB   8
002CC JBC_QUOTAS:
                .BLKB   66
0030E           .BLKB   2
00310 JBC_UIC:  .BLKB   4
00314 QUEUE_FAB:
                .BLKB   80
00364 QUEUE_RAB:
                .BLKB   68
003A8 QUEUE_NAM:
                .BLKB   96
00408 QUEUE_XAB:
                .BLKB   88
00460 QUEUE_RSA:
                .BLKB   255
0055F           .BLKB   1
00560 QUEUE_ALQ:
```

```
                        .BLKB   4
00564 QUEUE_MBF:
                        .BLKB   1
00565                   .BLKB   3
00568 ACCOUNTING_FABS:
                        .BLKB   8
00570 ACCOUNTING_RABS:
                        .BLKB   8
00578 ACCOUNT_FAB_A:
                        .BLKB   80
005C8 ACCOUNT_RAB_A:
                        .BLKB   68
0060C ACCOUNT_NAM_A:
                        .BLKB   96
0066C ACCOUNT_RSA_A:
                        .BLKB   255
0076B                   .BLKB   1
0076C ACCOUNT_FAB_B:
                        .BLKB   80
007BC ACCOUNT_RAB_B:
                        .BLKB   68
00800 ACCOUNT_NAM_B:
                        .BLKB   96
00860 ACCOUNT_RSA_B:
                        .BLKB   255
0095F                   .BLKB   1
00960 DIAG_FAB:
                        .BLKB   80
009B0 DIAG_RAB:
                        .BLKB   68
009F4 MBX_CHAN:
                        .BLKB   4
009F8 MBX_IOSB:
                        .BLKB   8
00A00 MBX_BUFFER:
                        .BLKB   1024
00E00 VALUE_STORAGE_BASE:
                        .BLKB   0
00E00 ITEM_PRESENT:
                        .BLKB   32
00E20 VALUE_GETQUI_BASE:
                        .BLKB   0
00E20 VALUE_ACCOUNTING_MESSAGE:
                        .BLKB   6
00E26 VALUE_ACCOUNTING_TYPES:
                        .BLKB   4
00E2A VALUE_AFTER_TIME:
                        .BLKB   8
00E32 VALUE_ALIGNMENT_PAGES:
                        .BLKB   1
00E33 VALUE_BASE_PRIORITY:
                        .BLKB   1
00E34 VALUE_BATCH_INPUT:
                        .BLKB   6
00E3A VALUE_BATCH_OUTPUT:
                        .BLKB   10
00E44 VALUE_BUFFER_COUNT:
```

```
                        .BLKB   1
00E45 VALUE_CHARACTERISTIC_NAME:
                        .BLKB   6
00E4B VALUE_CHARACTERISTIC_NUMBER:
                        .BLKB   1
00E4C VALUE_CHARACTERISTICS:
                        .BLKB   16
00E5C VALUE_CHECKPOINT_DATA:
                        .BLKB   6
00E62 VALUE_CLI:
                        .BLKB   6
00E68 VALUE_CPU_DEFAULT:
                        .BLKB   4
00E6C VALUE_CPU_LIMIT:
                        .BLKB   4
00E70 VALUE_DESTINATION_QUEUE:
                        .BLKB   8
00E78 VALUE_DEVICE_NAME:
                        .BLKB   6
00E7E VALUE_ENTRY_NUMBER:
                        .BLKB   4
00E82 VALUE_ENTRY_NUMBER_OUTPUT:
                        .BLKB   10
00E8C VALUE_EXTEND_QUANTITY:
                        .BLKB   2
00E8E VALUE_FILE_COPIES:
                        .BLKB   1
00E8F VALUE_FILE_IDENTIFICATION:
                        .BLKB   36
00EB3 VALUE_FILE_SETUP_MODULES:
                        .BLKB   6
00EB9 VALUE_FILE_SPECIFICATION:
                        .BLKB   6
00EBF VALUE_FIRST_PAGE:
                        .BLKB   4
00EC3 VALUE_FORM_DESCRIPTION:
                        .BLKB   6
00EC9 VALUE_FORM_LENGTH:
                        .BLKB   1
00ECA VALUE_FORM_MARGIN_BOTTOM:
                        .BLKB   1
00ECB VALUE_FORM_MARGIN_LEFT:
                        .BLKB   2
00ECD VALUE_FORM_MARGIN_RIGHT:
                        .BLKB   2
00ECF VALUE_FORM_MARGIN_TOP:
                        .BLKB   1
00ED0 VALUE_FORM_NAME:
                        .BLKB   6
00ED6 VALUE_FORM_NUMBER:
                        .BLKB   4
00EDA VALUE_FORM:
                        .BLKB   8
00EE2 VALUE_FORM_SETUP_MODULES:
                        .BLKB   6
00EE8 VALUE_FORM_STOCK:
                        .BLKB   6
```

```
00EEE VALUE_FORM_WIDTH:
                .BLKB   2
00EF0 VALUE_GENERIC_TARGET:
                .BLKB   996
012D4 VALUE_JOB_COPIES:
                .BLKB   1
012D5 VALUE_JOB_LIMIT:
                .BLKB   1
012D6 VALUE_JOB_NAME:
                .BLKB   6
012DC VALUE_JOB_RESET_MODULES:
                .BLKB   6
012E2 VALUE_JOB_SIZE_MAXIMUM:
                .BLKB   4
012E6 VALUE_JOB_SIZE_MINIMUM:
                .BLKB   4
012EA VALUE_JOB_STATUS_OUTPUT:
                .BLKB   10
012F4 VALUE_LAST_PAGE:
                .BLKB   4
012F8 VALUE_LIBRARY_SPECIFICATION:
                .BLKB   6
012FE VALUE_LOG_QUEUE:
                .BLKB   8
01306 VALUE_LOG_SPECIFICATION:
                .BLKB   6
0130C VALUE_NOTE:
                .BLKB   6
01312 VALUE_OPERATOR_REQUEST:
                .BLKB   6
01318 VALUE_OWNER_UIC:
                .BLKB   4
0131C VALUE_PAGE_SETUP_MODULES:
                .BLKB   6
01322 VALUE_PARAMETER_1:
                .BLKB   6
01328 VALUE_PARAMETER_2:
                .BLKB   6
0132E VALUE_PARAMETER_3:
                .BLKB   6
01334 VALUE_PARAMETER_4:
                .BLKB   6
0133A VALUE_PARAMETER_5:
                .BLKB   6
01340 VALUE_PARAMETER_6:
                .BLKB   6
01346 VALUE_PARAMETER_7:
                .BLKB   6
0134C VALUE_PARAMETER_8:
                .BLKB   6
01352 VALUE_PRIORITY:
                .BLKB   1
01353 VALUE_PROCESSOR:
                .BLKB   6
01359 VALUE_PROTECTION:
                .BLKB   4
0135D VALUE_QUEUE:
```

```
                        .BLKB   6
01363 VALUE_QUEUE_FILE_SPECIFICATION:
                        .BLKB   6
01369 VALUE_RELATIVE_PAGE:
                        .BLKB   4
0136D VALUE_RESERVED_INPUT_1:
                        .BLKB   1
0136E VALUE_RESERVED_INPUT_2:
                        .BLKB   2
01370 VALUE_RESERVED_INPUT_3:
                        .BLKB   4
01374 VALUE_RESERVED_INPUT_4:
                        .BLKB   6
0137A VALUE_RESERVED_OUTPUT_1:
                        .BLKB   10
01384 VALUE_RESERVED_OUTPUT_2:
                        .BLKB   10
0138E VALUE_SEARCH_STRING:
                        .BLKB   6
01394 VALUE_SCSNODE_NAME:
                        .BLKB   6
0139A VALUE_WSDEFAULT:
                        .BLKB   2
0139C VALUE_WSEXTENT:
                        .BLKB   2
0139E VALUE_WSQUOTA:
                        .BLKB   2
013A0 VALUE_STORAGE_END:
                        .BLKB   0

        JBC$_CLOSEOUT=      266328
        JBC$_NOCMKRNL=      272388
        JBC$_NOOPER=        272532
        JBC$_NOSYSNAM=      272404
        JBC$_OPENIN=        266392
        JBC$_OPENOUT=       266400
        JBC$_READERR=       266416
        JBC$_WRITEERR=      266448
                        .EXTRN  COMPLETE_JOB, COMPLETE_SRB_OUTPUT_ITEM
                        .EXTRN  CREATE_SRB, FETCH_VARIABLE_ITEM
                        .EXTRN  FETCH_VARIABLE_ITEM_LIST
                        .EXTRN  FIND_PENDING_JOBS
                        .EXTRN  FIND_PROCESS_DATA
                        .EXTRN  FLUSH_RECORD, LOCATE_SRB_OUTPUT_ITEM
                        .EXTRN  READ_RECORD, RELEASE_RECORD
                        .EXTRN  REWRITE_RECORD, SEND_SERVICE_RESPONSE_MESSAGE
                        .EXTRN  UPDATE_GETQUI_DATA

                        .PSECT  CODE,NOWRT,2

                 0FFC 00000     .ENTRY  SJC_BATCH_SERVICE, Save R2,R3,R4,R5,R6,R7,- ; 1158
                                        R8,R9,R10,R11                               ;
5E      FBF0    CE   9E 00002           MOVAB   -1040(SP), SP                       ;
50 00000000'    EF   D0 00007           MOVL    MBX, R0                             ; 1203
08          04  A0   E8 0000E           BLBS    4(R0), 1$                           ;
50  00042804    8F   D0 00012           MOVL    #272388, R0                         ; 1205
                     04 00019           RET                                         ;
```

```
                                7E  D4 0001A 1$:     CLRL    -(SP)                                   ; 1210
                            28  A0  DD 0001C         PUSHL   40(R0)                                  ; 1211
                                01  DD 0001F         PUSHL   #1                                      ; 1210
                00000000G   EF      03  FB 00021     CALLS   #3, FIND_PROCESS_DATA
                            08      50  E8 00028     BLBS    R0, 2$
                            50 00048040 8F  D0 0002B MOVL    #294976, R0                             ; 1214
                                04  00032           RET
                                5A  DD 00033 2$:     PUSHL   SMQ_N                                   ; 1219
                00000000G   EF      01  FB 00035     CALLS   #1, READ_RECORD
                            56      50  D0 0003C     MOVL    R0, SMQ
                                5B  DD 0003F         PUSHL   SJH_N                                   ; 1220
                00000000G   EF      01  FB 00041     CALLS   #1, READ_RECORD
                            5A      50  D0 00048     MOVL    R0, SJH
                            08  AE  D4 0004B         CLRL    FLAGS                                   ; 1225
             4D 00000000'   EF      02  E1 0004E     BBC     #2, ITEM_PRESENT+1, 8$                  ; 1226
                            59 00000000' EF D0 00056 MOVL    VALUE_BATCH_INPUT+2, P                  ; 1236
                            50 00000000' EF 3C 0005D MOVZWL  VALUE_BATCH_INPUT, R0                   ; 1237
                            58      FC A049 9E 00064 MOVAB   -4(R0)[P], P_END
                            58          59  D1 00069 3$:     CMPL    P, P_END                        ; 1242
                                35  1E 0006C         BGEQU   8$
                            50      02  A9 3C 0006E MOVZWL  2(P), TYPE                              ; 1251
                            57          89  3C 00072 MOVZWL  (P)+, SIZE                            ; 1252
                            59          02  C0 00075 ADDL2   #2, P                                 ; 1253
                 01 00008001    8F      50  CF 00078 CASEL   TYPE, #32769, #1                       ; 1258
                        0011            0006     00080 4$:   .WORD   5$-4$,-
                                                             6$-4$
                                1D  11 00084         BRB     8$                                      ; 1263
                            04          57  D1 00086 5$:     CMPL    SIZE, #4                        ; 1268
                                13  12 00089         BNEQ    7$
                        08  AE          69  D0 0008B MOVL    (P), FLAGS                              ; 1270
                                0D  11 0008F         BRB     7$                                      ; 1258
                            0C          57  D1 00091 6$:     CMPL    SIZE, #12                       ; 1276
                                08  1A 00094         BGTRU   7$
                00      69          57  2C 00096 MOVC5   SIZE, (P), #0, #12, 220(SJH)               ; 1281
                                00DC    CA      0009B
                            59          57  C0 0009E 7$:     ADDL2   SIZE, P                         ; 1290
                                C6  11 000A1         BRB     3$                                      ; 1242
                                10  AE  9F 000A3 8$:     PUSHAB  SRB                                 ; 1297
                    00000000G   EF      01  FB 000A6     CALLS   #1, CREATE_SRB
                                00000000' EF 9F 000AD    PUSHAB  VALUE_BATCH_OUTPUT                  ; 1298
                                0B  DD 000B3         PUSHL   #11
                                18  AE  9F 000B5         PUSHAB  SRB
                    00000000G   EF      03  FB 000B8     CALLS   #3, LOCATE_SRB_OUTPUT_ITEM
                            0C  AE      50  D0 000BF     MOVL    R0, DJI
                            53      0C  AE  D0 000C3     MOVL    DJI, DJIITM
                                03  12 000C7         BNEQ    9$                                      ; 1303
                                020E 31 000C9         BRW     33$
                            83 00030004 8F  D0 000CC 9$:     MOVL    #196612, (DJIITM)+              ; 1309
                            58          53  D0 000D3 MOVL    DJIITM, DJIFLG                          ; 1311
                                83  D4 000D6         CLRL    (DJIITM)+                               ; 1312
                            68      40  8F  88 000D8 BISB2   #64, (DJIFLG)                          ; 1314
                        04  AE      0C  AA  9E 000DC MOVAB   12(SJH), 4(SP)                          ; 1319
                03      04  BE          0E  E1 000E1 BBC     #14, @4(SP), 10$
                            68          10  88 000E6 BISB2   #16, (DJIFLG)
                03      11  AA          02  E1 000E9 10$:    BBC     #2, 17(SJH), 11$                ; 1320
                            68          20  88 000EE BISB2   #32, (DJIFLG)
                05      04  BE          0B  E1 000F1 11$:    BBC     #11, @4(SP), 12$                ; 1321
```

```
                        68              04  88 000F6          BISB2   #4, (DJIFLG)                                    ; 1323
                                        10  11 000F9          BRB     14$
                03      04      BE      0A  E1 000FB 12$:     BBC     #10, @4(SP), 13$                                ; 1326
                        68              02  88 00100          BISB2   #2, (DJIFLG)
                03      04      BE      0C  E1 00103 13$:     BBC     #12, @4(SP), 14$                                ; 1327
                        68              08  88 00108          BISB2   #8, (DJIFLG)
                                        53  DD 0010B 14$:     PUSHL   DJIITM                                          ; 1336
                                        0F  DD 0010D          PUSHL   #15                                             ; 1334
                                0180    CA  9F 0010F          PUSHAB  384(SJH)
                                        20  DD 00113          PUSHL   #32
                    00000000G   EF      04  FB 00115          CALLS   #4, FETCH_VARIABLE_ITEM
                        53              50  D0 0011C          MOVL    R0, DJIITM
                                        59  D4 0011F          CLRL    T                                               ; 1341
                        07      04      BE  E9 00121          BLBC    @4(SP), 15$                                     ; 1342
                        59      00E8    CA  D0 00125          MOVL    232(SJH), T
                                        09  11 0012A          BRB     16$
                04      0C      A6      02  E1 0012C 15$:     BBC     #2, 12(SMQ), 16$                                ; 1343
                        59      40      A6  D0 00131          MOVL    64(SMQ), T
                        57      0C      A6  9E 00135 16$:     MOVAB   12(SMQ), R7                                     ; 1344
                16      67              03  E1 00139          BBC     #3, (R7), 17$
                        68      80      8F  88 0013D          BISB2   #128, (DJIFLG)                                  ; 1347
                51      44      A6      01  C3 00141          SUBL3   #1, 68(SMQ), R1                                 ; 1348
                        50      FF      A9  9E 00146          MOVAB   -1(R9), R0
                        50              51  D1 0014A          CMPL    R1, R0
                                        04  1E 0014D          BGEQU   17$
                        59      44      A6  D0 0014F          MOVL    68(SMQ), T                                      ; 1350
                        08      04      BE  E8 00153 17$:     BLBS    @4(SP), 18$                                     ; 1352
                04      67              02  E0 00157          BBS     #2, (R7), 18$                                   ; 1353
                0A      67              03  E1 0015B          BBC     #3, (R7), 19$                                   ; 1354
                        83  00010004     8F  D0 0015F 18$:     MOVL    #65540, (DJIITM)+                               ; 1357
                        83              59  D0 00166          MOVL    T, (DJIITM)+                                    ; 1360
                        83      0108    CA  9B 00169 19$:     MOVZBW  264(SJH), (DJIITM)+                             ; 1367
                        83              04  B0 0016E          MOVW    #4, (DJIITM)+                                   ; 1368
                        50      0108    CA  9A 00171          MOVZBL  264(SJH), R0                                    ; 1371
                63      0109    CA      50  28 00176          MOVC3   R0, 265(SJH), (DJIITM)                          ; 1373
                        50      0104    CA  D0 0017C          MOVL    260(SJH), R0                                    ; 1378
                                        28  13 00181          BEQL    20$
                        6E              50  D0 00183          MOVL    R0, SMQ_N2                                      ; 1385
                                        50  DD 00186          PUSHL   R0
                    00000000G   EF      01  FB 00188          CALLS   #1, READ_RECORD
                        83      00B0    C0  9B 0018F          MOVZBW  176(SMQ_2), (DJIITM)+                           ; 1386
                        83              05  B0 00194          MOVW    #5, (DJIITM)+                                   ; 1387
                        51      00B0    C0  9A 00197          MOVZBL  176(SMQ_2), R1                                  ; 1390
                63      00B1    C0      51  28 0019C          MOVC3   R1, 177(SMQ_2), (DJIITM)                        ; 1392
                                        6E  DD 001A2          PUSHL   SMQ_N2                                          ; 1393
                    00000000G   EF      01  FB 001A4          CALLS   #1, RELEASE_RECORD
                                        53  DD 001AB 20$:     PUSHL   DJIITM                                          ; 1402
                                        06  DD 001AD          PUSHL   #6                                              ; 1400
                                01A0    CA  9F 001AF          PUSHAB  416(SJH)
                                        06  DD 001B3          PUSHL   #6
                    00000000G   EF      04  FB 001B5          CALLS   #4, FETCH_VARIABLE_ITEM
                        53              50  D0 001BC          MOVL    R0, DJIITM
                                        53  DD 001BF          PUSHL   DJIITM                                          ; 1410
                                        07  DD 001C1          PUSHL   #7                                              ; 1408
                                01B2    CA  9F 001C3          PUSHAB  434(SJH)
                                        20  DD 001C7          PUSHL   #32
                    00000000G   EF      04  FB 001C9          CALLS   #4, FETCH_VARIABLE_ITEM_LIST
```

```
                                53              50  D0 001D0          MOVL    R0, DJIITM                                      ;
                            83  0010000C       8F  D0 001D3          MOVL    #1048588, (DJIITM)+                             ; 1415
                    63      0148            CA  0C  28 001DA          MOVC3   #12, 328(SJH), (DJIITM)                         ; 1421
                                59              01  CE 001E0          MNEGL   #1, T                                           ; 1426
                    09                      67  17  E1 001E3          BBC     #23, (R7), 21$                                  ; 1427
                        01                  A8  01  88 001E7          BISB2   #1, 1(DJIFLG)                                   ; 1430
                                59      010E    C6  3C 001EB          MOVZWL  270(SMQ), T                                     ; 1431
                0E          04              BE  12  E1 001F0  21$:    BBC     #18, @4(SP), 22$                                ; 1433
        59      0172        CA              10  00  ED 001F5          CMPZV   #0, #16, 370(SJH), T                            ; 1436
                                            05  1E 001FC              BGEQU   22$                                             ;
                                59      0172    CA  3C 001FE          MOVZWL  370(SJH), T                                     ;
                                            59  D5 00203  22$:        TSTL    T                                               ; 1438
                                            0A  19 00205              BLSS    23$                                             ;
                            83  00110004        8F  D0 00207          MOVL    #1114116, (DJIITM)+                             ; 1441
                            83                  59  D0 0020E          MOVL    T, (DJIITM)+                                    ; 1444
                                59              01  CE 00211  23$:    MNEGL   #1, T                                           ; 1451
                                09      03      A7  E9 00214          BLBC    3(R7), 24$                                      ; 1452
                        01                  A8  02  88 00218          BISB2   #2, 1(DJIFLG)                                   ; 1455
                                59      0110    C6  3C 0021C          MOVZWL  272(SMQ), T                                     ; 1456
                0E          04              BE  13  E1 00221  24$:    BBC     #19, @4(SP), 25$                                ; 1458
        59      0174        CA              10  00  ED 00226          CMPZV   #0, #16, 372(SJH), T                            ; 1461
                                            05  1E 0022D              BGEQU   25$                                             ;
                                59      0174    CA  3C 0022F          MOVZWL  372(SJH), T                                     ;
                                            59  D5 00234  25$:        TSTL    T                                               ; 1463
                                            0A  19 00236              BLSS    26$                                             ;
                            83  00120004        8F  D0 00238          MOVL    #1179652, (DJIITM)+                             ; 1466
                            83                  59  D0 0023F          MOVL    T, (DJIITM)+                                    ; 1469
                                59              01  CE 00242  26$:    MNEGL   #1, T                                           ; 1476
                    09                      67  19  E1 00245          BBC     #25, (R7), 27$                                  ; 1477
                        01                  A8  04  88 00249          BISB2   #4, 1(DJIFLG)                                   ; 1480
                                59      0112    C6  3C 0024D          MOVZWL  274(SMQ), T                                     ; 1481
                0E          04              BE  14  E1 00252  27$:    BBC     #20, @4(SP), 28$                                ; 1483
        59      0176        CA              10  00  ED 00257          CMPZV   #0, #16, 374(SJH), T                            ; 1486
                                            05  1E 0025E              BGEQU   28$                                             ;
                                59      0176    CA  3C 00260          MOVZWL  374(SJH), T                                     ;
                                            59  D5 00265  28$:        TSTL    T                                               ; 1488
                                            0A  19 00267              BLSS    29$                                             ;
                            83  00130004        8F  D0 00269          MOVL    #1245188, (DJIITM)+                             ; 1491
                            83                  59  D0 0027C          MOVL    T, (DJIITM)+                                    ; 1494
                                52      08      AE  E8 00273  29$:    BLBS    FLAGS, 32$                                      ; 1499
                                52      00F0    CA  D0 00277          MOVL    240(SJH), R2                                    ; 1505
                                            07  12 0027C              BNEQ    30$                                             ;
                                56      00F4    CA  D0 0027E          MOVL    244(SJH), SQR_N                                 ; 1507
                                            18  11 00283              BRB     31$                                             ;
                                            52  DD 00285  30$:        PUSHL   R2                                              ; 1510
                    00000000G               EF  01  FB 00287          CALLS   #1, READ_RECORD                                 ;
                                54              50  D0 0028E          MOVL    R0, SQR                                         ;
                                56              64  D0 00291          MOVL    (SQR), SQR_N                                    ; 1511
                                            52  DD 00294              PUSHL   R2                                              ; 1512
                    00000000G               EF  01  FB 00296          CALLS   #1, RELEASE_RECORD                              ;
                        00F0                CA  56  D0 0029D  31$:    MOVL    SQR_N, 240(SJH)                                 ; 1518
                                            25  13 002A2              BEQL    32$                                             ; 1523
                                            56  DD 002A4              PUSHL   SQR_N                                           ; 1529
                    00000000G               EF  01  FB 002A6          CALLS   #1, READ_RECORD                                 ;
                                54              50  D0 002AD          MOVL    R0, SQR                                         ;
                                68      40      8F  8A 002B0          BICB2   #64, (DJIFLG)                                   ; 1534
                            83  0002001C        8F  D0 002B4          MOVL    #131100, (DJIITM)+                              ; 1539
```

```
                63       1C   A4              1C  28 002BB          MOVC3   #28, 28(SQR), (DJIITM)                  ; 1545
                                              56  DD 002C0          PUSHL   SQR_N                                   ; 1548
                   00000000G  EF              01  FB 002C2          CALLS   #1, RELEASE_RECORD                      ;
                                              83  D4 002C9 32$:     CLRL    (DJIITM)+                               ; 1555
                7E             53        0C   AE  C3 002CB          SUBL3   DJI, DJIITM, -(SP)                      ; 1562
                                         14   AE  9F 002D0          PUSHAB  SRB                                     ; 1560
                   00000000G  EF              02  FB 002D3          CALLS   #2, COMPLETE_SRB_OUTPUT_ITEM            ;
                                              5B  DD 002DA 33$:     PUSHL   SJH_N                                   ; 1568
                   00000000G  EF              01  FB 002DC          CALLS   #1, REWRITE_RECORD                      ;
                                              01  DD 002E3          PUSHL   #1                                      ; 1574
                                         14   AE  9F 002E5          PUSHAB  SRB                                     ;
                   00000000G  EF              02  FB 002E8          CALLS   #2, SEND_SERVICE_RESPONSE_MESSAGE       ;
                                              50  D4 002EF          CLRL    R0                                      ; 1576
                                                  04 002F1          RET                                             ;

; Routine Size:  754 bytes,    Routine Base:  CODE + 0000
```

```
;  540          1577  1 GLOBAL ROUTINE BATCH_DELETION(SMQ_N,SJH_N): NOVALUE=
;  541          1578  1
;  542          1579  1 !++
;  543          1580  1 !
;  544          1581  1 ! FUNCTIONAL DESCRIPTION:
;  545          1582  1 !       This routine handles the deletion of a batch process.
;  546          1583  1 !
;  547          1584  1 ! INPUT PARAMETERS:
;  548          1585  1 !       SMQ_N           - Record number of SMQ.
;  549          1586  1 !       SJH_N           - Record number of SJH.
;  550          1587  1 !
;  551          1588  1 ! IMPLICIT INPUTS:
;  552          1589  1 !       MBX             - Pointer to buffered mailbox message.
;  553          1590  1 !
;  554          1591  1 ! OUTPUT PARAMETERS:
;  555          1592  1 !       NONE
;  556          1593  1 !
;  557          1594  1 ! IMPLICIT OUTPUTS:
;  558          1595  1 !       NONE
;  559          1596  1 !
;  560          1597  1 ! ROUTINE VALUE:
;  561          1598  1 !       NONE
;  562          1599  1 !
;  563          1600  1 ! SIDE EFFECTS:
;  564          1601  1 !       NONE
;  565          1602  1 !
;  566          1603  1 !--
;  567          1604  1
;  568          1605  2 BEGIN
;  569          1606  2 LOCAL
;  570          1607  2         FLUSH_SMQ,                      ! Flag indicating SMQ should be flushed
;  571          1608  2         SMQ:            REF BBLOCK,     ! Pointer to SMQ
;  572          1609  2         SJH:            REF BBLOCK,     ! Pointer to SJH
;  573          1610  2         SJH_NT,                         ! Record number of tentative SJH
;  574          1611  2         SJH_NP,                         ! Record number of predecessor of SJH
;  575          1612  2         SJH_P:          REF BBLOCK;     ! Pointer to predecessor of SJH
;  576          1613  2
;  577          1614  2
;  578          1615  2 ! Read and update the queue header.
;  579          1616  2 !
;  580          1617  2 SMQ = READ_RECORD(.SMQ_N);
;  581          1618  2 SMQ[SMQ$B_CURRENT_JOB_COUNT] = .SMQ[SMQ$B_CURRENT_JOB_COUNT] - 1;
;  582          1619  2 QUEUE_REFERENCE_COUNT = .QUEUE_REFERENCE_COUNT - 1;
;  583          1620  2 FLUSH_SMQ = FALSE;
;  584          1621  2
;  585          1622  2
;  586          1623  2 ! Search the current queue for the job record.
;  587          1624  2 !
;  588          1625  2 SJH_NP = .SMQ_N;
;  589          1626  2 SJH_NT = .SMQ[SMQ$L_CURRENT_LIST];
;  590          1627  2 WHILE .SJH_NT NEQ 0 DO
;  591          1628  3     BEGIN
;  592          1629  3     SJH = READ_RECORD(.SJH_NT);
;  593          1630  3     IF .SJH_NT EQL .SJH_N
;  594          1631  3     THEN
;  595          1632  4         BEGIN
;  596          1633  4
```

```
;  597          1634  4          ! Unlink the job from the current queue.
;  598          1635  4          !
;  599          1636  4          UPDATE_GETQUI_DATA(.SJH_N, .SJH);
;  600          1637  4          IF .SJH_NP EQL .SMQ_N
;  601          1638  4          THEN
;  602          1639  5              BEGIN
;  603          1640  5              SMQ[SMQ$L_CURRENT_LIST] = .SJH[SYM$L_LINK];
;  604          1641  5              IF .SJH[SYM$L_LINK] EQL 0 THEN SMQ[SMQ$L_CURRENT_LIST_END] = 0;
;  605          1642  5              FLUSH_SMQ = TRUE;
;  606          1643  5              END
;  607          1644  4          ELSE
;  608          1645  5              BEGIN
;  609          1646  5              SJH_P[SYM$L_LINK] = .SJH[SYM$L_LINK];
;  610          1647  5              IF .SJH[SYM$L_LINK] EQL 0
;  611          1648  5              THEN
;  612          1649  6                  BEGIN
;  613          1650  6                  SMQ[SMQ$L_CURRENT_LIST_END] = .SJH_NP;
;  614          1651  6                  FLUSH_SMQ = TRUE;
;  615          1652  5                  END;
;  616          1653  5              REWRITE_RECORD(.SJH_NP);
;  617          1654  4              END;
;  618          1655  4
;  619          1656  4
;  620          1657  4          ! If the SMQ is dirty and needs to be re-written before doing
;  621          1658  4          ! COMPLETE_JOB, do so.  Then re-read it for subsequent processing.
;  622          1659  4          !
;  623          1660  4          IF .FLUSH_SMQ
;  624          1661  4          THEN
;  625          1662  4              FLUSH_RECORD(.SMQ_N);
;  626          1663  4
;  627          1664  4          ! Complete the job.
;  628          1665  4          !
;  629          1666  4          COMPLETE_JOB(.SJH_N, .SJH, .SMQ, .MBX);
;  630          1667  4
;  631          1668  4
;  632          1669  4          ! Find more work for the queue.
;  633          1670  4          !
;  634          1671  4          FIND_PENDING_JOBS(.SMQ_N, .SMQ);
;  635          1672  4          ! (Note: probably need only to RELEASE here, not REWRITE.)
;  636          1673  4          REWRITE_RECORD(.SMQ_N);
;  637          1674  4          RETURN;
;  638          1675  3          END;
;  639          1676  3
;  640          1677  3
;  641          1678  3      ! Advance to next job.
;  642          1679  3      !
;  643          1680  3      IF .SJH_NP NEQ .SMQ_N THEN RELEASE_RECORD(.SJH_NP);
;  644          1681  3      SJH_NP = .SJH_NT;
;  645          1682  3      SJH_P = .SJH;
;  646          1683  3      SJH_NT = .SJH[SYM$L_LINK];
;  647          1684  2      END;
;  648          1685  1 END;
; INFO#250                  L1:1646
; Referenced LOCAL symbol SJH_P is probably not initialized
```

```
                                07FC 00000          .ENTRY  BATCH_DELETION, Save R2,R3,R4,R5,R6,R7,R8,-      ; 1577
                                                            R9,R10
                    5A 00000000G   EF  9E 00002     MOVAB   READ_RECORD, R10                                ;
                    59 00000000G   EF  9E 00009     MOVAB   REWRITE_RECORD, R9                              ;
                    56       04    AC  D0 00010     MOVL    SMQ_N, R6                                       ; 1617
                                   56  DD 00014     PUSHL   R6                                              ;
                    6A             01  FB 00016     CALLS   #1, READ_RECORD                                 ;
                    52             50  D0 00019     MOVL    R0, SMQ                                         ;
                           0115    C2  97 0001C     DECB    277(SMQ)                                        ; 1618
                       00000000'   EF  D7 00020     DECL    QUEUE_REFERENCE_COUNT                           ; 1619
                                   57  D4 00026     CLRL    FLUSH_SMQ                                       ; 1620
                    54             56  D0 00028     MOVL    R6, SJH_NP                                      ; 1625
                    55       48    A2  D0 0002B     MOVL    72(SMQ), SJH_NT                                 ; 1626
                                   01  12 0002F 1$:  BNEQ    2$                                              ; 1627
                                       04 00031     RET                                                     ;
                                   55  DD 00032 2$:  PUSHL   SJH_NT                                          ; 1629
                    6A             01  FB 00034     CALLS   #1, READ_RECORD                                 ;
                    53             50  D0 00037     MOVL    R0, SJH                                         ;
                 08 AC             55  D1 0003A     CMPL    SJH_NT, SJH_N                                   ; 1630
                                   61  12 0003E     BNEQ    8$                                              ;
                                   53  DD 00040     PUSHL   SJH                                             ; 1636
                          08       AC  DD 00042     PUSHL   SJH_N                                           ;
              00000000G EF         02  FB 00045     CALLS   #2, UPDATE_GETQUI_DATA                          ;
                    56             54  D1 0004C     CMPL    SJH_NP, R6                                      ; 1637
                                   0E  12 0004F     BNEQ    4$                                              ;
                 48 A2             63  D0 00051     MOVL    (SJH), 72(SMQ)                                  ; 1640
                                   03  12 00055     BNEQ    3$                                              ; 1641
                          4C       A2  D4 00057     CLRL    76(SMQ)                                         ;
                    57             01  D0 0005A 3$:  MOVL    #1, FLUSH_SMQ                                   ; 1642
                                   11  11 0005D     BRB     6$                                              ; 1637
                    68             63  D0 0005F 4$:  MOVL    (SJH), (SJH_P)                                  ; 1646
                                   07  12 00062     BNEQ    5$                                              ; 1647
                 4C A2             54  D0 00064     MOVL    SJH_NP, 76(SMQ)                                 ; 1650
                    57             01  D0 00068     MOVL    #1, FLUSH_SMQ                                   ; 1651
                                   54  DD 0006B 5$:  PUSHL   SJH_NP                                          ; 1653
                    69             01  FB 0006D     CALLS   #1, REWRITE_RECORD                              ;
                    09             57  E9 00070 6$:  BLBC    FLUSH_SMQ, 7$                                   ; 1660
                                   56  DD 00073     PUSHL   R6                                              ; 1662
              00000000G EF         01  FB 00075     CALLS   #1, FLUSH_RECORD                                ;
                       00000000'   EF  DD 0007C 7$:  PUSHL   MBX                                             ; 1666
                                   52  DD 00082     PUSHL   SMQ                                             ;
                                   53  DD 00084     PUSHL   SJH                                             ;
                          08       AC  DD 00086     PUSHL   SJH_N                                           ;
              00000000G EF         04  FB 00089     CALLS   #4, COMPLETE_JOB                                ;
                                   52  DD 00090     PUSHL   SMQ                                             ; 1671
                                   56  DD 00092     PUSHL   R6                                              ;
              00000000G EF         02  FB 00094     CALLS   #2, FIND_PENDING_JOBS                           ;
                                   56  DD 0009B     PUSHL   R6                                              ; 1673
                    69             01  FB 0009D     CALLS   #1, REWRITE_RECORD                              ;
                                       04 000A0     RET                                                     ; 1632
                    56             54  D1 000A1 8$:  CMPL    SJH_NP, R6                                      ; 1680
                                   09  13 000A4     BEQL    9$                                              ;
                                   54  DD 000A6     PUSHL   SJH_NP                                          ;
              00000000G EF         01  FB 000A8     CALLS   #1, RELEASE_RECORD                              ;
                    54             55  D0 000AF 9$:  MOVL    SJH_NT, SJH_NP                                  ; 1681
```

```
                58              53  D0 000B2        MOVL    SJH, SJH_P                              ; 1682
                55              63  D0 000B5        MOVL    (SJH), SJH_NT                           ; 1683
                              FF74  31 000B8        BRW     1$                                      ; 1627
                                    04 000BB        RET                                             ; 1685

; Routine Size:  188 bytes,    Routine Base:  CODE + 02F2
```

```
;  650          1686  1 END
;  651          1687  0 ELUDOM
```

```
;                                  PSECT SUMMARY
;
;       Name                      Bytes                          Attributes
;
;  COMMON                          5024  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  OVR,NOPIC,ALIGN(2)
;  CODE                             942  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
```

```
;                               Library Statistics
;
;                                       -------- Symbols --------      Pages       Processing
;       File                            Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]LIB.L32;1         18619        69         0      1000          00:01.3
```

```
; Information:  1
; Warnings:     0
; Errors:       0
```

```
;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:BATCH/OBJ=OBJ$:BATCH MSRC$:BATCH/UPDATE=(ENH$:BATCH)

; Size:          942 code + 5024 data bytes
; Run Time:        00:21.8
; Elapsed Time:    02:37.1
; Lines/CPU Min:    4634
; Lexemes/CPU-Min: 40032
; Memory Used:  420 pages
; Compilation Complete
```

BATCH
LIS

BROADCAST
LIS

BUFFERS
LIS

CONTROL
LIS

ASYNCHRON
LIS

CHECKPROT
LIS